

PhD Dissertations and Master's Theses

5-2021

Data-Driven Architecture to Increase Resilience In Multi-Agent Coordinated Missions

D. F. Embry-Riddle Aeronautical University

Follow this and additional works at: https://commons.erau.edu/edt

Part of the Aerospace Engineering Commons

Scholarly Commons Citation

F., D., "Data-Driven Architecture to Increase Resilience In Multi-Agent Coordinated Missions" (2021). *PhD Dissertations and Master's Theses.* 634. https://commons.erau.edu/edt/634

This Dissertation - Open Access is brought to you for free and open access by Scholarly Commons. It has been accepted for inclusion in PhD Dissertations and Master's Theses by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.

DATA-DRIVEN ARCHITECTURE TO INCREASE RESILIENCE IN MULTI-AGENT COORDINATED MISSIONS

By

D. F.

A Dissertation Submitted to the Faculty of Embry-Riddle Aeronautical University In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Aerospace Engineering

> May 2021 Embry-Riddle Aeronautical University Daytona Beach, Florida

DATA-DRIVEN ARCHITECTURE TO INCREASE RESILIENCE IN MULTI-AGENT COORDINATED MISSIONS

By

D. F.

This Dissertation was prepared under the direction of the candidate's Dissertation Committee Chair, Dr. Hever Moncayo, Department of Aerospace Engineering, and has been approved by the members of the Dissertation Committee. It was submitted to the Office of the Senior Vice President for Academic Affairs and Provost, and was accepted in the partial fulfillment of the requirements for the Degree of Doctor of Philosophy in Aerospace Engineering.

DISSERTATION COMMITTEE

Chairman, Dr. Hever Moncayo	Member, Dr. Richard Prazenica	
Member, Dr. Morad Nazari	Member, Dr. Eric Coyle	
Graduate Program Coordinator, Dr. Sirish Namilae	Date	
Dean of the College of Engineering, Dr. Maj Mirmirani	Date	
Senior Vice President for Academic Affairs and Provost, Dr. Lon Moeller	Date	

ACKNOWLEDGMENTS

Thank you to the Chair of the Aerospace Engineering Department, to my advisor, to my advisory committee, and to all in the Aerospace Engineering Department. Receive my gratitude and appreciation for you all. Thank you.

ABSTRACT

The rise in the use of Multi-Agent Systems (MASs) in unpredictable and changing environments has created the need for intelligent algorithms to increase their autonomy, safety and performance in the event of disturbances and threats. MASs are attractive for their flexibility, which also makes them prone to threats that may result from hardware failures (actuators, sensors, onboard computer, power source) and operational abnormal conditions (weather, GPS denied location, cyber-attacks). This dissertation presents research on a bio-inspired approach for resilience augmentation in MASs in the presence of disturbances and threats such as communication link and stealthy zero-dynamics attacks. An adaptive bio-inspired architecture is developed for distributed consensus algorithms to increase fault-tolerance in a network of multiple high-order nonlinear systems under directed fixed topologies. In similarity with the natural organisms' ability to recognize and remember specific pathogens to generate its immunity, the immunity-based architecture consists of a Distributed Model-Reference Adaptive Control (DMRAC) with an Artificial Immune System (AIS) adaptation law integrated within a consensus protocol. Feedback linearization is used to modify the high-order nonlinear model into four decoupled linear subsystems. A stability proof of the adaptation law is conducted using Lyapunov methods and Jordan decomposition. The DMRAC is proven to be stable in the presence of external time-varying bounded disturbances and the tracking error trajectories are shown to be bounded. The effectiveness of the proposed architecture is examined through numerical simulations. The proposed controller successfully ensures that consensus is achieved among all agents while the adaptive law

simultaneously rejects the disturbances in the agent and its neighbors. The architecture also includes a health management system to detect faulty agents within the global network. Further numerical simulations successfully test and show that the Global Health Monitoring (GHM) does effectively detect faults within the network.

TABLE OF CONTENTS

AC	CKNC	DWLEDGMENTS	iii
AE	BSTR	ACT	iv
LI	ST O	F FIGURES	viii
LIS	ST O	F TABLES	xi
N(OMEI	NCLATURE	xii
SY	YMB(DLS	xiv
1.	Intro	oduction	1
	1.1.	Motivation	1
	1.2.	Problem Statement	4
	1.3.	Contributions	5
	1.4.	Dissertation Outline	6
2.	Lite	rature Review	9
	2.1.	Multi-Agent System Networks	9
		2.1.1. Consensus in Multi-Agent Systems	12
	2.2.	Disturbance Rejection and Fault-Tolerant Control in MAS	14
	2.3.	Immunity-Inspired Control Laws	16
	2.4.	Adaptive Control Laws	18
		2.4.1. Adaptive Control Applications in MAS	20
	2.5.	Cyber-Physical Attacks in MASs	22
		2.5.1. Communication Link Attack	22
		2.5.2. Zero-Dynamics Attack	23
3.	Mat	hematical Background	26
	3.1.	Graph Theory	26
	3.2.	Matrix Theory	30
		3.2.1. Eigenvalues and Eigenvectors	31
		3.2.2. Jordan Matrix Decomposition of the Laplacian Matrix	31
		3.2.3. Jordan Matrix Decomposition Example	32
		3.2.4. Kronecker Product	34
	3.3.	Lyapunov Stability	34
	3.4.	Consensus Protocols in Networked Multi-Agent Systems	37
		3.4.1. First and Second-Order Consensus Algorithms	37
		3.4.2. High-Order Consensus Algorithms	39
		3.4.3. Consensus with Reference Tracking	40
	3.5.	Quadrotor Dynamics	41
	3.6.	Feedback Linearization for Quadrotor Dynamics	44
		- v	

4.	Imm	unity-Inspired Distributed MRAC
	4.1.	Intelligent Systems
	4.2.	Immunity-Inspired Architecture
		4.2.1. Local Self
		$4.2.2. Global Self \dots \dots$
	4.3.	Immunity-Inspired Adaptation Law
	4.4.	Distributed Consensus Architecture
		4.4.1. Quadrotor Dynamics
		4.4.2. Reference Model
		4.4.3. Baseline Control Law
		4.4.4. Adaptive Control Law
		4.4.5. Proposed DMRAC
	4.5.	Distributed MRAC for Formation Control
5.	Stab	ility Analysis
C	DM	
6.	DMI	RAC Numerical Simulations
	6.1.	Bounded Disturbances
		6.1.1. Consensus Control
	0.0	$6.1.2. Formation Control \dots 78$
	6.2.	Communication Link Attack
		$6.2.1. Consensus Control \dots 92$
		6.2.2. Formation Control
	6.3.	Zero-Dynamics Attack
		$6.3.1. Consensus Control \dots 104$
7	Clah	al Solf Numerical Simulations
1.	G100	Connection of Solf and Non Solf Humananaaa
	(.1.	Generation of Self and Non-Self Typerspace 100 7.1.1 Self /Nen Self Depresentation 107
		$7.1.1. \text{Self/Non-Self Representation} \dots \dots$
		$(1.1.2. Hierachical Multiself Strategy \dots \dots$
	7.0	$(1.1.3. \text{ Antibody Generation} \dots \dots$
	7.2.	Fault Detection in Quadrotor MAS
		$\begin{array}{cccccccccccccccccccccccccccccccccccc$
		7.2.2. High-Order Detectors
		7.2.3. Detection Rates and False Alarms
		7.2.4. Global Detection of Faulty Agent
0	Dian	ussion and Conclusion 121
0.		$ \begin{array}{llllllllllllllllllllllllllllllllllll$
	0.1.	Future work on Distributed MRAC $\dots \dots \dots$
	8.2.	Future work on Global Health Monitoring
RI	EFER	ENCES
AI	PPEN	DIX A
AI	PPEN	DIX B

LIST OF FIGURES

Figure Page		Page
2.1	Network types	11
2.2	Model-reference adaptive control bloc-diagram representation \ldots .	19
2.3	Adaptive control timeline	21
2.4	ZDA bloc diagram	24
3.1	Examples of graph representation	27
3.2	Different communication topologies for three agents	30
3.3	Topology for the Jordan decomposition example	33
3.4	Stability concept	36
3.5	Quadrotor forces and moments	41
4.1	Interaction of an agent within its network	52
4.2	Humoral feedback mechanism	54
4.3	T_s -cells nonlinear function	55
4.4	Consensus control for a multi-agent system with immune adaptive augmentation	64
5.1	System adaptive function	71
6.1	Information exchange topology	75
6.2	Agents reaching an agreement on position	77
6.3	Reference model: nominal conditions for DMRAC performance	79
6.4	Consensus without adaptation (baseline controller) \ldots	80
6.5	Consensus with adaptation (baseline & adaptive controller) \ldots	81

Figure

6.6	Evolution of Agent 1 states under bounded disturbance with and without AIS-inspired adaptation law	82
6.7	Evolution of Agent 2 states under bounded disturbance with and without AIS inspired adaptation law	83
6.8	Evolution of Agent 3 states under bounded disturbance with and without AIS inspired adaptation law	84
6.9	Evolution of Agent 4 states under bounded disturbance with and without AIS inspired adaptation law	85
6.10	Evolution of the adaptive inputs for Agent 1 under bounded disturbance with and without AIS inspired adaptation law	86
6.11	Evolution of the adaptive inputs for Agent 2 under bounded disturbance with and without AIS inspired adaptation law	87
6.12	Evolution of the adaptive inputs for Agent 3 under bounded disturbance with and without AIS inspired adaptation law	88
6.13	Evolution of the adaptive inputs for Agent 4 under bounded disturbance with and without AIS inspired adaptation law	89
6.14	Formation control subject to a small disturbance	90
6.15	Formation control subject to a large disturbance	91
6.16	3D position of agents' trajectories after communication link attack $% \mathcal{L}^{(1)}$.	93
6.17	Agent states reaching agreement after a communication link attack on Agent 4	95
6.18	Agent states reaching agreement after communication link attack $\ .$.	96
6.19	Effect of communication link attack on formation keeping with undi- rected topology when Agent 1 states are under attack	97
6.19	(cont.) Effect of communication link attack on formation keeping with undirected topology when Agent 1 states are under attack	98

Figure		Page
6.20	Effect of communication link attack on formation with directed topology when an agent is under attack.	99
6.20	(cont.) Effect of communication link attack on formation with directed topology when an agent is under attack.	100
6.21	Example of an unbounded disturbance generated for a zero-dynamics attack	101
6.22	ZDA injection in simulation environment	103
6.23	Zero-dynamics attack: Continuous vs Discrete time	104
7.1	2D projection using x_2 vs ϕ_3	116
7.2	2D projection using x_2 vs x_3	117
7.3	AB activation with fault in Agent 1	120
7.4	New communication topology for a formation keeping case \ldots .	121
7.5	Fault detection in Agent 1 by the ABs in Self 10	122
7.6	Fault detection in Agent 1 by the AB in Self 5	123
7.7	Fault detection in Agent 1 by the ABs in Self 9	124
7.8	DR & FA in Selves $1 - 12$	125
7.9	3D representation of Self 11 (per Table 7.5)	128
7.10	Fault detection results in Self 1 and Self 2	129
8.1	DR and FA after disturbance injection in an agent.	155

LIST OF TABLES

Table		Page
6.1	Initial conditions	76
6.2	Adaptive controller gain values	76
7.1	Main biological terms	111
7.2	List of all considered 2D self projections	115
7.3	Generated high-order selves (part a)	118
7.4	Generated high-order selves (part b)	122
7.5	Selves for global fault detection in Agent 1	126
8.1	Detection rate and false alarms	153

NOMENCLATURE

$n\mathbf{D}$	<i>n</i> -dimensional
AC	Abnormal Condition
ADCL	Advanced Dynamics and Control Laboratory
AIS	Artificial Immune System
CPS	Cyber-Physical Systems
DMRAC	Distributed Model-Reference Adaptive Control
DoS	Denial-of-Service
DR	Detection Rate
ERAU	Embry-Riddle Aeronautical University
FA	False Alarm
FDI	Failure Detection and Identification
GHM	Global Health Monitoring
GPS	Global Positioning System
HM	Health Monitoring
HMS	Hierarchical Multi-Self Strategy
НО	High-Order
MAS	Multi-Agent System
MRAC	Model-Reference Adaptive Control
NASA	National Aeronautics & Space Administration

- **NLDI** Non-Linear Dynamic Inversion
- **NN** Neural Networks
- **NS** Negative Selection
- **RDSU** Raw Data Set Union
- **UAV** Unmanned Aerial Vehicle
- **ZDA** Zero-Dynamics Attack

SYMBOLS

\otimes	Kronecker product
$\lambda_i(A)$	i^{th} eigenvalue of matrix A
A > 0	A is a positive definite matrix
1_p	$p \times 1$ column vector of all ones
sup	supremum, the least upper bound
inf	infinimum, the greatest lower bound
I_n	$n \times n$ identity matrix
n	number of agents
\mathcal{A}	adjacency matrix
\mathcal{D}	degree matrix
L	Laplacian matrix
\mathcal{N}_i	neighbor set of agent i
$J_k(\lambda)$	Jordan block
x, y, z	position in Earth reference frame
x_r, y_r	desired 2D position
$\phi, \; \theta, \; \psi$	Euler angles
m	quadrotor mass
l	length from center of mass of quadrotor to the rotor
F	forces
M	moments

K	drag coefficients
Ι	moments of inertia
δ	bounded, time-varying disturbance
x	state vector
\mathbf{x}_m	reference model state vector
u'	virtual control input
u_l	local controller
u_c	consensus controller
u_b	baseline controller
u_{AD}	adaptive controller
e	error
K_1	local controller gain
K_2	consensus controller gain
K_{AD}	adaptive gain
G_{ZDA}	ZDA transfer function
$a^{ji}_{\mathbf{x}}$	communication link attack signal between nodes \boldsymbol{i}
H_{∞}	H_{∞} control
\mathcal{L}_1	$\mathcal{L}_1 ext{ control}$

1. Introduction

The onset of technological advances and the continuous development of computation, actuation, and sensing devices have accelerated research on cooperative control across industries. Successful engineering tests in this field have enabled the deployment of multiple agents for various missions. This has turned out favorable in cost effectiveness over using single agents.

A Multi-Agent System (MAS) is comprised of multiple autonomous subsystems interacting with each other via a communication network. These subsystems, also referred to as agents, can be robots, vehicles, sensors, or process plants that work cooperatively to achieve certain tasks. Compared to the single-agent control, the cooperative control of multiple agents has advantages such as larger redundancy, higher robustness, and greater fault-tolerance. The main advantages of these distributed systems include reduced costs in their design, manufacturing, and operation with control algorithms designed to provide scalability and robustness during missions (Y. Cao et al., 2013; Ren & Beard, 2008).

1.1. Motivation

Several mission applications that involve the deployment of groups of autonomous vehicles demand decentralized swarming capabilities and require advanced and novel technologies to increase overall mission performance, particularly if they are operating under complex and dynamically changing environments. Aerospace MAS applications include formation control, attitude alignment, rendezvous, coordinated decision making and flocking. These advanced autonomous systems will require adequate intelligent systems to increase mission safety and optimize performance within complex, unstructured and dynamic operating environments.

Research today is largely exploring ways of increasing autonomy of aerial systems by providing a level of intelligence to guarantee desired behaviors on a mission. Biological mechanisms such as the immune system exhibit robust, adaptive, and distributed cognitive capabilities and thus present a possible solution to the problem. In a MAS network, these technologies are expected to increase autonomy by maintaining control of the agent during unrecoverable conditions and other uncertainties such as faults and failures of hardware (actuators, sensors, onboard computer, power source) as well as exogenous disturbances (weather, GPS denied location, physical obstacles, cyber-attacks or radar areas). The process can then be followed by intelligent decision making to determine any required modifications to the mission such that the best possible scenario is achieved (Moncayo et al., 2011).

In this dissertation, a novel implementation of biologically inspired algorithms to effectively increase MAS resilience during a mission is investigated. Within this configuration, each agent's vulnerability may cause new concerns arising from the agents' interactions within the system. Therefore, the impact of unpredictable threats on the system can be avoided or minimized if assessed and processed in time. The proposed approach leads to an efficient technique to ensure dynamic resilience of MAS networks where the adaptive system works in parallel with a baseline controller. The outputs from the adaptive system are added to the baseline controls to achieve required tracking.

This study introduces an architecture based on the adaptive function of the

immune system and considers four scenarios: consensus and tracking, consensus and formation, communication link attack and Zero-Dynamics Attack (ZDA). Concepts borrowed from the biological immune system response, graph theory, optimization, as well as the global/local environments are developed herein and their respective interactions are applied to the proposed controller which in turn ensures consensus among the agents while rejecting bounded disturbances.

The study also explores the challenge of coordinating the agents for successful mission protection. Each agent within the system is aware of its environment and of the MAS behavior; it has its own independence yet cooperates with the other agents, as modeled in the immune system configuration. The communication between the agents is implicit and the algorithm can be used to synchronize how multiple agents in distributed networks communicate and coordinate to accomplish specific mission objectives by reaching an agreement on certain states, despite limited computational resources and sensing capabilities. Moreover, agents have their own local perspective of the whole system. The nature of these control algorithms provide both scalability and robustness in response to changes in a dynamic environment.

Unmanned Aerial Vehicles (UAVs) are preferred in MAS as they provide unique capabilities for various mission objectives, including surveillance, mapping, target detection, and environmental monitoring. In this work, distributed consensus problems are applied to a network of multiple quadrotors. The focus herein is for the multiple quadrotor system to achieve and maintain consensus under fixed topologies while rejecting disturbances. Nonlinear dynamics are common to the agent's model in many consensus problems under complex networks. Given the difficulty in design of consensus algorithms with nonlinear dynamics, feedback linearization is used to transform the high-order nonlinear quadrotor model into a group of four linear subsystems (Wang et al., 2013).

1.2. Problem Statement

Existing approaches to enhance MAS coordination are being directed towards providing the needed autonomy and usability of these systems. However, theoretical formulations and specialized architectures are still needed to increase resilience of a MAS under different types of threats that can degrade overall mission performance. Fault-tolerant control for MAS has drawn particular attention and thereby some theoretical research works, yet has seen limited applications to specific vehicle dynamics. This work provides alternative tools for the design of advanced algorithms to increase resilience of MAS missions operating under nominal conditions or in the presence of internal or exogenous disturbances.

Research Objective

The overall research objective is to develop a distributed bio-inspired adaptive consensus architecture to increase resilience in a network of multiple agents under both bounded and unbounded disturbances. The scope of this objective states how to address the following questions :

- What is the state-of-the-art? (Chapter 2),
- What is the proposed framework? (Chapters 3 & 4),
- How can the proposed framework be applied to a MAS under disturbances? (Chapters 4 & 5),
- How does the proposed architecture compare to a baseline framework in terms

of robustness, adaptability and consensus performance under disturbances? (Chapter 6).

1.3. Contributions

The contribution of this work is to introduce the design and implementation of a novel consensus algorithm based on biologically inspired adaptive controllers for MAS networks. To this end, an Artificial Immune System (AIS) architecture is developed through the application of bio-inspired algorithms. The specific algorithms provide agents with fault rejection capabilities that enable them to quickly adapt to rapidly changing environments. In order for a fault to be assessed and processed in time by the system, a Failure Detection and Identification (FDI) strategy is required both within the formation and for each individual agent. In this study, a fault-tolerant distributed consensus controller is applied to a networked quadrotor system. MAS theories are combined with fault-tolerance principles to design a resilient multi-agent system.

This study builds on previous and current research on applying AIS in a MAS by further looking into the design of a resilient control law based on consensus algorithms. The novelty of this research effort resides on the following elements:

- Development of a distributed model-reference adaptive controller with an immune-inspired adaptation law for disturbance rejection.
- Application of this new controller to a networked system for both consensus and tracking problems. This controller is tested in simulation against sinusoidal disturbances.
- Application of this new controller to the detection and compensation of

cyber-attacks, namely communication link attacks and zero-dynamics attacks.

- Stability analysis of the proposed controller using Lyapunov method and graph theoretic properties. It is proven that with the DMRAC with AIS augmentation, the tracking error trajectories of the system in the presence of a bounded disturbance are indeed bounded, and the system is stable.
- Development of a Global Health Monitoring (GHM) architecture based on previous Advanced Dynamics and Control Laboratory (ADCL) findings using the AIS paradigm. The interactions between the agents as well as the fault detection process used are defined within this architecture.

The impact and significance of this study relies on mission applications that involve the deployment of groups of autonomous vehicles with decentralized swarming capabilities and require advanced and novel technologies to increase overall mission performance while operating in complex and changing environments. Simulations were carried out at the ADCL at Embry-Riddle Aeronautical University (ERAU) to develop and operate representative scale prototypes in the development of new algorithms.

1.4. Dissertation Outline

This dissertation is composed of eight chapters as presented below. The chapters are then followed by references, appendices and the list of publications.

Chapter 1 lays out the research motivation. The problem statement is formulated and presented along with the research relevance, the contributions and the general objectives.

In Chapter 2, a detailed literature review on consensus protocols and disturbance

rejection in MAS along with a recapitulation of existing control methods in MAS and an overview of the literature on cyber attacks is presented.

Chapter 3 defines the theoretical background for the study. Basic algebraic graph theory, required mathematical concepts such as matrix theory and Lyapunov stability theory are introduced along with a summary on consensus algorithms. Linearization of the quadrotor dynamics, the agent of choice in this work, is also detailed.

A comprehensive description of the distributed MRAC architecture is provided in Chapter 4. It includes a presentation of the various components of the DMRAC, from the reference model to the immunity-inspired adaptation law, as well as the proposed consensus controller.

The proposed controller is then proven to be stable in the presence of external time-varying bounded disturbances as demonstrated in Chapter 5, where a Lyapunov-inspired function is derived. The Laplacian matrix is then transformed into the real Jordan form in order to conduct the global stability analysis in the framework of Lyapunov functions. Conditions are derived for the MAS to guarantee global consensus.

Simulation results of the controller under case scenarios are presented in Chapter 6. Results are shown for both consensus and formation control under bounded disturbances and cyber-attacks, namely communication link attack and zero-dynamics attack. The performance of the DMRAC is compared to that of the baseline controller in all results.

Chapter 7 introduces, illustrates and analyzes the global health monitoring (or management) system through multiple simulation results. After an in-depth presentation of the GHM system, the given results demonstrate that the GHM system is able to detect faults within the MAS when a sinusoidal bounded disturbance has occurred.

Finally, the conclusion in Chapter 8 summarizes and analyzes the findings in this work. A scope for future work is discussed, giving possible extensions to the research.

2. Literature Review

In this chapter, the context of the dissertation is provided in a review of the related literature relevant to the researched topics. This chapter discusses the theories, developments and popular methods across industries but mainly pertinent to engineering applications. It outlines some of the works on multi-agent systems, distributed model-reference adaptive control, consensus control and artificial immune systems.

The literature includes the work of pioneering scholars as (Forrest et al., 1994; Parker, 1993; Morse, 1980; Takahashi & Yamada, 1998; Kaufman et al., 1997; Dasgupta, 1999), influential book authors as Ren & Beard (2008) and Mesbahi & Egerstedt (2010), and a panorama of thorough studies presented in academic papers as (Olfati-Saber & Murray, 2004; C. Cao & Hovakimyan, 2007; Moncayo et al., 2011; Yucelen & Egerstedt, 2012). These cited works present their research goals, methods and analysis and, for a majority, the authors arrived at similar conclusions mainly using numerical tests. With the extensive interest in MAS, several authors have taken the task to summarize the trends in survey articles referenced throughout the chapter (Yang et al., 2020; W. Wang et al., 2020; Tahir et al., 2019; Zuo et al., 2018; Gulzar et al., 2018; Ding et al., 2018; Campion et al., 2018; Qin et al., 2017; Arfat & Eassa, 2016; Oh et al., 2015; Yang et al., 2014; Y. Cao et al., 2013; Dressler & Akan, 2010). An overview of MASs precedes specific literature subject content.

2.1. Multi-Agent System Networks

A multi-agent system is formed by a number of agents connected together to achieve a desired goal specified by the design (Arfat & Eassa, 2016). Exploring various ways to use multi-agent systems is now a widely sought after cross-industry research field of interest. The intelligence of MASs can vary in the method, function, procedure, approach and algorithmic setup. Multi-agent systems are also favorable in solving difficult problems that could be impossible for an individual agent to handle. In any operation or mission, a MAS requires certain important aspects, such as co-ordination, co-operation, negotiation and communication, to achieve fault tolerant control capabilities.

As mentioned here above, agents in MAS work on behalf of a user to accomplish defined goals. For effective cooperative functioning in groups through local communication, the control of networked multi-agent systems has been the focus of increased research activity (Olfati-Saber et al., 2007; Ren & Beard, 2008). The aim is for the networked system to function autonomously and bypass collaboration with humans for repetitive, risky, and often critical missions.

Centralized, Decentralized and Distributed Networks: Currently, the trend is to replace single vehicles by multiple yet simpler vehicles that add flexibility and robustness to the network (Y. Cao et al., 2013). A *centralized* control system has each component of the system dependent on a central controller for its operation outcome based on the assumption that a central station, or leader, has availability and power to control a whole group of vehicles. In contrast, *decentralized* control systems have multiple leaders and work locally with direct communication to the agents. A network is commonly referred to as *distributed* when each component in the system contributes to the global network based on its own local information, therefore eliminating all centralization. This information may lead to achieve



Figure 2.1 Network types

common group objectives, relative position information or common control algorithms. Figure 2.1 illustrates the differences between the types of networks.

Fixed vs. Switching Topologies: Multi-agent systems operate within either a fixed or a switching communication topology. A fixed communication topology means that the communication patterns between agents, whether in directed or undirected graphs, does not vary during the entirety of the mission. In switching topologies, the communication patterns of the MAS shift at specific times (Fattahi & Afshar, 2019; Mao et al., 2020). In practical applications of MAS tracking systems, communication topologies between agents are time-varying since the original tracking agents could be replaced by others during the mission, for example. Since dynamical behaviors of MASs are subject to not only agent dynamics but also communication topology, it is sometimes not feasible for the agents to maintain a fixed communication topology due to various reasons such as collision avoidance, communication link failure or communication range limitations. Therefore, in the event of such a scenario, it is mandatory to consider switching topology (D. Xue et al., 2013; Ajwad et al., 2021).

While in operation within a distributed network, the multi-agent system is susceptible to fail due to lack of resources, a highly possible occurrence. The resources for MAS may not be available due to either an agent failure, machine crashes, process failure, software failure, cyber-attacks, communication failure and/or hardware failure. Therefore, many researchers have proposed fault tolerant approaches to overcome these threats (Y. Cao et al., 2013).

The main objective of the MAS is to enable a network of agents to perform tasks collaboratively with limited and selected communication. Agents operate in consensus protocols via distributed controllers equipped with local sensor information. In a distributed MAS, an agent's local dynamics impact the global coordination when experiencing a disturbance. Thus, at the local agent level, an adaptive controller is preferred for handling a MAS prone to disturbances.

Control algorithms for networked multi-agent systems are generally computed distributively without having a centralized entity monitoring the activity of agents. As a result, unforeseen adverse conditions such as uncertainties or attacks to the communication network and/or failure of agent components can easily result in system instability and prohibit the accomplishment of system-level objectives (F. & Moncayo, 2020).

2.1.1. Consensus in Multi-Agent Systems

In recent years, MAS consensus has continued to draw increasing interest for several applications (Ding et al., 2018; Yang et al., 2020; Zuo et al., 2018). Similar to other aspects in MAS, consensus algorithms have drawn great research interests mainly for their multiple uses in aerospace applications, namely spacecraft formation flying, sensor networks, unmanned air vehicle formations, to name a few.

Consensus serves as a fundamental principle in the design of distributed multi-agent coordination algorithms. As a typical collective behavior in a network of autonomous agents, consensus requires that the agents in the MAS reach an agreement on specifically selected points of interest. Studying consensus involves considering the main features that characterize systems, such as actuation, control, communication, computation, and vehicle dynamics. In a typical MAS, many agents are grouped to form a cooperative system, in which each agent shares the local information only with its neighboring agents under a distributed structure to achieve a special global common behavior in a cooperative way.

Early works focused on MAS consensus using integer-order dynamics, also referred to as first-order agent dynamics (Olfati-Saber & Murray, 2004). Since then, a number of studies on applications using second-order agent dynamics and higher-order agent dynamics have been carried out (Mei et al., 2016; Yu et al., 2013; Ren et al., 2007; H. Zhang & Lewis, 2012; Z. Li et al., 2014).

Other studies have also included consensus control with fractional-order dynamics (Bai et al., 2018; Gong, 2016; W. Zhu et al., 2017). Fractional-order systems stem from traditional integer-order systems, their distinction being in the memory term that provides infinite memory and hereditary properties (Podlubny, 1998). Other research efforts include consensus tracking control, also known as leader-following consensus control, consensus-based containment control (Lui et al., 2021; Yuan et al., 2019) and consensus-based formation control (Oh & Ahn, 2011; Oh et al., 2015; Kuriki & Namerikawa, 2013; Z. Yang et al., 2019).

Another proposed approach for a fully distributed consensus protocol is by adopting an adaptive gain updating scheme (Mei et al., 2016). A number of scholars referenced herein have documented results on this design for MAS with undirected topologies. Authors Zhang & Lewis (2012) and Peng et al. (2013) report that when unknown nonlinearities can only be estimated over a compact set, Neural Networks (NN) have also been used for compensation given their approximation capability.

2.2. Disturbance Rejection and Fault-Tolerant Control in MAS

Research on MAS has created a pool of literature surveys on related topics. These include adaptive control, distributed multi-agent coordination, and consensus control (Gong et al., 2020; Ding et al., 2018). Existing literature on disturbance rejection and fault-tolerant control are summarized in this section.

Significant research activities have been conducted in the design and analysis of fault diagnosis and accommodation schemes. Most of these methods utilize a centralized architecture, where the diagnostic module is designed based on a global mathematical model of the overall system and is required to have real-time access to all sensor measurements (Blanke et al., 2016). Centralized methods present limitations in large computational resources and communication overhead, and are not suitable for many application domains with large networked systems. As a result, there is a growing tendency to research on the development of distributed fault diagnosis schemes for multi-agent systems (Keliris et al., 2013; Ferrari et al., 2012; Shames et al., 2011).

In Davoodi et al. (2018), the authors formulate and present a fault detection

controller for problems within the domains of linear time-invariant systems, the Markovian jump and MAS. The presented Markovian jump can be controlled to detect and isolate single or multiple faults in the system. They also present a mixed H_{∞} formulation using distributed detection filters based on relative output information between the agents.

In their research, authors Sun et al. (2016) study the observer-based consensus disturbance rejection with dynamic coupling. The authors design a distributed adaptive observer to decouple the adaptive coupling gain from the control input. They add a low-pass filter to reduce the initial adaptive rate of the coupling gain. The advantage is that the high-gain coupling has no direct impact on the magnitude of the control input.

For fault tolerance in MAS, (O'Keeffe et al., 2018; Yucelen & Egerstedt, 2012; Mange, 2013; J. Sun et al., 2016; Ismail & Timmis, 2010) each present methodologies in the fields of fault detection diagnosis and recovery. Their respective methods are implemented on modeling disturbances within nonlinear optimization problems.

Subsystem failures on an agent may affect the global performance and lead to instability. Fault tolerant consensus has been reported in literature with protocols and architectures that take into account actuator bias (S. Chen et al., 2015). Most existing consensus protocols such as the ones reported in (Bai et al., 2018; Gong, 2016; Weng et al., 2014; Yu et al., 2013; H. Zhang & Lewis, 2012; W. Zhu et al., 2017) require a level of global information to fully characterize the health of the network, hence the importance of a fully distributed consensus control approach that is independent of the need for global information. In (Lafferriere et al., 2005; Yucelen & Egerstedt, 2012; De La Torre & Yucelen, 2018), the authors present methodologies for adaptive architectures with implementation in vehicle formations and decentralized swarms. They also discuss nonlinear interconnected systems and the effect of persistent disturbances.

2.3. Immunity-Inspired Control Laws

The AIS metaphor has been applied successfully to a variety of problems ranging from anomaly detection and pattern recognition, to data mining and computer security (Castro & Zuben, 2002; Seresht & Azmi, 2014). In aerospace systems, the application of the AIS paradigm to fault detection was pioneered by Dasgupta (1999) and Kumar (2003). Authors Takahashi & Yamada (1998) formulated an adaptive controller that mimics the interaction of T-cells. Research efforts have used the AIS paradigm integrated with a Hierarchical Multi-Self (HMS) strategy to perform failure detection, identification, and evaluation of aerospace systems (Moncayo et al., 2011, 2012; Perhinschi et al., 2013).

In line with the trend in technology, future missions will involve both manned and unmanned aircraft. These missions will require methods that enable unmanned aircraft with intelligent manoeuvring capabilities. For aerospace applications, bio-inspired techniques are in the spotlight of international artificial intelligence.

The biological immune system protects the body against intruders by recognizing and destroying harmful cells or molecules. It can be matched to a robust adaptive system that can tackle various disturbances and uncertainties. Another critical aspect of the immune system is that it can remember how previous encounters were successfully eliminated. Similarly, an AIS can respond faster to similar encounters in the future for coordination, self healing and path planning, among other applications.

Effective fault-tolerant schemes in research now include a continuous interest in mimicking the natural immune system against faults in MAS. In these studies, the MAS is represented as the biological system, the sensor network, or the robotic team. Multi-agent cooperative operation strategies have been proposed based on the immune network theory. AIS tools can be used effectively together to solve complex engineering problems including fault tolerance. One application of AIS algorithms are Intrusion Detection Systems (IDS). These mechanisms attempt to discover abnormal access to computers by analyzing various interactions for detecting abnormal behaviors before they cause widespread damage to the system (Yang et al., 2014).

Negative Selection: In the application of AIS, the Negative Selection (NS) algorithm proposed by Forrest et al. (1994) pioneered the algorithms that have since been adapted for the generation of immune detectors. Initially set up to detect data manipulation caused by a virus in a computer, it is inspired by the generation of T-cells in the immune system. The Real-valued Negative Selection Algorithm (RNSA) presented by González et al. (2003) studied the detectors and antigens in the real-value space. The V-Detector algorithm proposed by Ji & Dasgupta (2009) widens the detection areas by turning the fixed-length detectors in RNSA into the variable-sized detectors.

Further in the use of AIS in MAS control, Timmis et al. (2016) present their findings on artificial immune systems and Salehi & Selamat (2011) present a hybrid simple artificial immune system for implementation on negative selection. These methods find their application in equilibrium-based strategy algorithms, with the main challenge being recovery from failure mode. Mange (2013) proposes a methodology for negative selection AIS and a normal three-parameter particle swarm optimization algorithm to be implemented in optimization tasks. The NS algorithm is popular for data representation as it sets limits for matching rules, for the detector generation mechanism, and the detection process. The common data to be processed include numerical, categorical, boolean and textual data, and the representations mostly used are either string or real-value vector representations (Clotet et al., 2018).

2.4. Adaptive Control Laws

Adaptive control is a technique that is commonly used for adjusting the parameters of a plant in real-time to maintain a desired level of performance when the parameters of the system are unknown and/or change with time. The Model-Reference Adaptive Control (MRAC) offers an approach for the solution of problems related to adaptive control in various applications (Nguyen, 2018). By creating a closed-loop controller, the MRAC tries to compare the output of the plant with a standard reference response and various parameters of the plant change with this response, as shown in Figure 2.2. Adaptive controllers are categorized as either direct or indirect based on their implementation or adaptation law. The direct adaptive controller will directly estimate the parameters used in the controller, whereas the indirect adaptive controller will estimate the parameters used to calculate the gains. Most common methods include the model-reference adaptive control, direct adaptive control, neural networks, and model identification adaptive controllers. The evolution of adaptive control laws is presented in Figure 2.3.

According to Rohrs et al. (1985), a traditional adaptive control design is not



Figure 2.2 Model-reference adaptive control bloc-diagram representation

robust enough when faced with bounded disturbances. To further increase robustness in the adaptive controller, researchers have proposed certain specific modifications, such as σ -modification, *e*-modification and AIS-augmentation (A. E. Rocha, 2016). For the stability and asymptotic tracking in the absence of disturbances, the most common adaptive control schemes are either Lyapunov-based or estimation-based schemes (Krstic et al., 1995). On the one hand, the Lyapunov-based scheme derives its adaptive laws from a Lyapunov stability analysis. On the other hand, the estimation-based scheme is selected from gradient or least-squares optimization algorithms. Research reveals that the initial stability analysis of the MRAC was designed and analyzed via Lyapunov theory and has been referred to as a pioneer scheme in adaptive controls (Morse, 1980).

One more candidate for controlling nonlinear unknown systems are the NN based adaptive control methodologies with a universal function approximation ability that guarantees closed-loop performance. In this case, it is still difficult to apply to time-varying systems, as the desired weights of the NN become time-varying variables (C. Cao & Hovakimyan, 2007).

Adaptive control design is in general limited by the unbalanced performance between fast adaptation and robustness. This is caused by the high frequencies in the control signals and makes it more sensitive to time delays. In Hovakimyan & Cao (2010), an \mathcal{L}_1 adaptive control theory is used to decouple the fast adaptation from robustness. Figure 2.3 traces the evolution in the types of adaptive controllers. The presented controllers at each point in time have been tested by NASA during multiple test flights since the 1950's. In their application, adaptive control laws are yet to be certified for aerospace vehicles.

2.4.1. Adaptive Control Applications in MAS

Research to date proposes different types of adaptive control architectures that have been used to increase robustness in MAS. Examples of these approaches are described below. Previous works in distributed control of MAS used fixed-gain control strategies which are not capable of recovering the desired system performance in the presence of unpredicted changing environments. This is specifically due to the fact that control algorithms for these systems are generally computed distributively without having a centralized entity monitoring the activity of agents.

Adverse conditions such as uncertainties or attacks to the communication network and/or failure of certain agent components can easily result in system instability and prohibit the accomplishment of system-level objectives (Bullo et al., 2009; Shamma, 2007). In a more recent study, the considered class of adverse conditions can be mitigated by the proposed adaptive control approach, even if all


Figure 2.3 Adaptive control timeline (adapted from Nguyen (2018)). This is the evolution in the types of adaptive controllers. Although these controllers have been tested by NASA during multiple test flights since the 1950's, adaptive control laws are yet to be certified for aerospace vehicles.

agents face disturbances (De La Torre & Yucelen, 2018).

Author Luo (2013) presents a control framework for distributed multi-agent coordination with unknown nonlinear uncertainties by integrating \mathcal{L}_1 adaptive control and cooperative control laws. The \mathcal{L}_1 adaptive control law is used to handle the mismatched dynamics between the real agent's and the ideal agent's dynamics, which mainly stem from unknown nonlinear uncertainties.

In their study, Peng et al. (2013) created a distributed model-reference adaptive control (DMRAC) architecture for cooperative tracking of unpredictable and changing MAS, where a reference model is used as a virtual group leader. The authors design two adaptive laws, one adjusting the coupling weights and another adjusting the neural network weights, both based on the neighbor's relative state information. The controller guarantees synchronizing each agent to the reference model's past signals, and any undirected connected communication graph in the closed-loop network is uniformly ultimately bounded. Furthermore, the controller can be implemented in a fully distributed manner by each agent without relying on information from the global network. As opposed to most neural networks cooperative tracking controllers, where bounded tracking error results are obtained, this paper reported asymptotic tracking performance by the controller.

2.5. Cyber-Physical Attacks in MASs

With the fast advance of electronics, high-speed computing techniques, and communications, the control of Cyber-Physical Systems (CPSs) has become a highly active research discipline for its monumental applications, including power grid systems and public health systems (Tahoun & Arafa, 2021; Feng et al., 2017). The exponential increase in privacy violations and security attacks experienced in CPSs has led to a wide interest in studying false data-injection attacks on MAS. For example, an AIS framework proposed by Tarao & Okamoto (2016) protects servers on the Internet against cyber attacks and examines detection performance using simulated machine learning techniques. In this section, two types of cyber-attacks on MAS are presented: the communication link attack and the zero-dynamics attack.

2.5.1. Communication Link Attack

Cyber-physical systems that have no protection in the transfer of data between sensors, controllers and actuators can easily become vulnerable prey to attackers. This can be both at the local and global network levels. Communication link attacks can be classified as cyber-attacks on networks that are subjected to false data-injection. The two most common types of communication attacks are attacks on the link among the sensors, the controllers and the actuators of the local networked agent level and attacks on the link among the agents and their neighbors at the global network level (Tahoun & Arafa, 2021).

The different types of cyber attacks have heavily exposed CPSs to certain specific categories of attacks such as Denial-of-Service (DoS), false data-injection and replay cyber-attacks. In DoS cyber-attacks, the attackers attempt to cut off the connections between the different parts of the system. The attackers in the false data-injection cyber-attacks aim to replace the original packets with false data packets when they are transferred between sensors, controllers and actuators or among agents via a communication network. The attackers in the replay cyber attacks aim to deceive the receiver as having received correct data which in reality is repeated or delayed data, making this type of attack undetectable (Pasqualetti et al., 2012a, 2012b; Teixeira et al., 2015). These attacks can have a high impact on the tracking behaviors of the MAS. Any failure in one local agent can spread to its neighboring agents resulting in a ripple degradation of the entire network system.

2.5.2. Zero-Dynamics Attack

The Zero-Dynamics Attack (ZDA) is classified among stealthy attacks known for being a security challenge as it hides its attack signal in the null-space of the state-space representation of the targeted control system (Figure 2.4). In so doing, the attack is not detectable via conventional detection methods that are framed for observation. The ZDA can be used to confound the network controller to accept false



Figure 2.4 ZDA bloc diagram (Kim & Shim, 2019)

data, thus exposing the system to the attacker's desired state of instability by maliciously changing the system dynamics via this stealthy topology attack.

Recent research in this area include Mao et al. (2019) who propose a multi-rate \mathcal{L}_1 adaptive controller to detect the ZDA in sampled data control systems by removing certain unstable zeros of discrete-time systems. Earlier work on ZDA defense strategies proposed limiting assumptions regarding the connectivity of network topology and the number of agents under attack. The detection worked for a single faulty agent in second-order systems and it was deduced that the defense strategy cannot detect ZDA in multi-agent system setup. Teixeira et al. (2015) went on to report that the defender can use certain changes in system dynamics to detect the ZDA, on condition that the defender is informed of the attack start time (Teixeira et al., 2015; Mao et al., 2019; Kim et al., 2020).

Besides component failures, cyber-physical systems are prone to malignant attacks, and specific analysis tools as well as monitoring mechanisms can be developed to enforce system security and reliability. In their study, Li et al. (2020) replicated the stealthy attack to guarantee consensus in the MAS, and proposed an event-triggered mechanism to control the MAS. The goal in this dissertation is not to detect the attacks but to specifically evaluate whether the proposed control architecture can slow down the effect of the ZDA on the agent dynamics so that it, the ZDA, can be detected by a global-self health monitoring system.

Summary: A review of existing methods for control of multi-agent systems was presented in this chapter. Starting with an overview of MAS, the literature content included centralized and decentralized MAS, consensus in MAS, disturbance rejection and fault-tolerant control in MAS, immunity-inspired control laws, adaptive control laws, adaptive control applications in MAS and cyber-physical attacks in MASs.

3. Mathematical Background

The mathematical foundation of the dissertation is provided in this chapter. Starting with a review of graph theory, matrix theory and Lyapunov stability, consensus protocols and high-order consensus algorithms are presented. This is followed by a presentation of linear and nonlinear equations required to model high-order systems. The following notations will be used throughout this chapter. $\lambda_i(A)$ represents the i^{th} eigenvalue of matrix A. A > 0 means that A is a positive definite matrix. \otimes denotes the Kronecker product.

3.1. Graph Theory

Graph theory is applied to multi-agent systems to mathematically represent the dynamic interactions within a network. A multi-agent network can be described as a graph, where agents are the nodes of the graph and an edge between two nodes represents their ability to communicate (Mesbahi & Egerstedt, 2010). The interaction between agents through exchanging information is modeled by directed or undirected graphs, as shown in Figure 3.1.

Definition 3.1 (Directed graph). A directed graph (or digraph) is a pair $\mathcal{G} = (V, E)$ defined by a set $V = \{1, \ldots, n\}$ of nodes (or vertices) and a set $E \subset V \times V$ of edges.

The set of neighbors of node *i* is denoted by $\mathcal{N}_i = \{j \mid (i, j) \in E\}$. $j \in \mathcal{N}_i$ indicates that agent *i* can sense (receive information from) agent *j*. As shown in Figure 3.1b, the direction of information flow between two agents is suitably given by an edge's orientation.



Figure 3.1 Examples of graph representation (Mesbahi & Egerstedt, 2010). The communication between two nodes (v) in an undirected graph is bidirectional. In a directed graph, the communication is given by the edge's (e) orientation.

The adjacency matrix $\mathcal{A} = [a_{ij}] \in \mathbb{R}^{n \times n}$ of \mathcal{G} given by,

$$a_{ij} = \begin{cases} 1, & \text{if } (i,j) \in E; \\ 0, & \text{otherwise,} \end{cases}$$
(3.1)

is a condensed representation of the interactions in a graph containing n vertices.

The degree matrix $\mathcal{D} \in \mathbb{R}^{n \times n}$ of \mathcal{G} is the diagonal matrix containing the vertex-degrees of \mathcal{G} on the diagonal, that is,

$$\mathcal{D} \triangleq \operatorname{diag}(d_1, \dots, d_n), \tag{3.2}$$

where d_i is the in-degree of a vertex (node) *i*. The in-degree of a node *i* is given by the number of its neighbors,

$$d_i = \sum_{j \in \mathcal{N}_i} a_{ij} \ . \tag{3.3}$$

The Laplacian matrix of a graph, $\mathcal{L} \in \mathbb{R}^{n \times n}$, given by $\mathcal{L} = \mathcal{D} - \mathcal{A}$, plays a

The principal conditions of \mathcal{L} that provide a general understanding of its importance are listed as follows:

- Symmetry: For an undirected graph, *L* is symmetric. This is not the case for directed graphs.
- Eigenvalues: Let $\lambda_i(\mathcal{L})$ be the i^{th} eigenvalue of \mathcal{L} with

 $\lambda_1(\mathcal{L}) \leq \lambda_2(\mathcal{L}) \leq \cdots \leq \lambda_n(\mathcal{L})$. Since \mathcal{L} has zero row sums, $\lambda_1(\mathcal{L}) = 0$ is an eigenvalue of \mathcal{L} with an associated eigenvector $\mathbf{1}$, where $\mathbf{1} \triangleq [1, \dots, 1]^T$ is an $n \times 1$ column vector of ones.

 Connectivity: λ₂(L) is the algebraic connectivity, which is positive if and only if the undirected graph is connected. The algebraic connectivity quantifies the convergence rate of consensus algorithms.

Definition 3.2 (Spanning tree). If the graph has a spanning tree, there is a non-repeated eigenvalue at $\lambda_1 = 0$ and all other eigenvalues have positive real parts (Ren & Beard, 2008).

Theorem 3.1. The graph \mathcal{G} is connected if and only if $\lambda_2(\mathcal{G}) > 0$ (Mesbahi & Egerstedt, 2010).

In a decentralized network of multiple agents, task accomplishment depends on the connectivity measure, i.e. how well the agents can communicate with each other (M. Ji & Egerstedt, 2007). The connectivity could be impacted if agents are lost during the mission (Yucelen & Egerstedt, 2012).

Continuing on the example in Equation 3.4 and Equation 3.5, the eigenvalues of \mathcal{L} for the graph in Figure 3.1a are [0, 0.83, 2.69, 4, 4.48] and $\lambda_2 = 0.83 > 0$. This graph is connected and therefore stable. On the other hand, the eigenvalues of \mathcal{L} for the graph in Figure 3.1b are [2, 0, 1, 0] and $\lambda_2 = 0$. This graph is disconnected and therefore unstable. It is important to note that this is due to the connectivity of the graph and not whether its topology is directed or undirected. Figure 3.2 shows three different directed communication topologies where in all cases the graph is connected and the corresponding system is stable.

For an undirected graph, \mathcal{L} is symmetric positive semi-definite. However, \mathcal{L} for a directed graph does not have this property. In both cases, 0 is a simple eigenvalue of \mathcal{L} if and only if the graph has a directed spanning tree. All of the nonzero eigenvalues



Figure 3.2 Different communication topologies for three agents (Ren & Beard, 2008)

of \mathcal{L} for digraphs have positive real parts and can be located using Geršgorin's disc theorem.

Theorem 3.2 (Geršgorin's disk theorem). All eigenvalues of a matrix $E = [e_{ij}] \in \mathbb{R}^{N \times N}$ are located within the union of N disks,

$$\bigcup_{i=1}^{N} \left\{ \lambda \in \mathcal{C} \mid |\lambda - e_{ij}| \le \sum_{j \ne i} |e_{ij}| \right\}.$$
(3.6)

The Geršgorin disks for the Laplacian matrix \mathcal{L} are centered at the in-degrees d_i and have radius equal to d_i . Let d_{max} be the largest in-degree of \mathcal{G} . The largest Geršgorin disk of \mathcal{L} is given by a circle \mathcal{C} centered at d_{max} and of radius of d_{max} . If the graph has a spanning tree, all nonzero eigenvalues have positive real parts and are within \mathcal{C} (Lewis et al., 2014).

3.2. Matrix Theory

To further detail the application of the matrices presented in the previous section, a selective review of matrix theory is given. Eigenvalues and eigenvectors, the Jordan matrix decomposition method and the Kronecker product are put forward.

3.2.1. Eigenvalues and Eigenvectors

Given a square matrix $A \in \mathbb{R}^{n \times n}$, $\lambda \in \mathbb{C}$ is called an eigenvalue of A if there exists a nonzero *n*-dimensional column vector $v \in \mathbb{R}^{n \times 1}$ such that,

$$Av = \lambda v. \tag{3.7}$$

The eigenvalues of A are defined as the solutions of,

$$\det(\lambda I_n - A) = 0. \tag{3.8}$$

A vector v satisfying Equation 3.8 is called an eigenvector of A corresponding to eigenvalue λ .

3.2.2. Jordan Matrix Decomposition of the Laplacian Matrix

Lemma 3.1 (Ding, 2014). For a Laplacian matrix that satisfies $\mathcal{L}\mathbf{1} = 0$, where $\mathbf{1} = [1, \dots, 1]^T \in \mathbb{R}^n$, there exists a similarity transformation T with the first column of $T, T_1 = \mathbf{1}$, such that,

$$T^{-1}\mathcal{L}T = J, \tag{3.9}$$

with J being a block diagonal matrix in the real Jordan form,

where $J \in \mathbb{R}^{n_k}$ for k = 1, ..., p are the Jordan blocks for real eigenvalues $\lambda_k > 0$ with

the multiplicity n_k in the form,

$$J_{k} = \begin{bmatrix} \lambda_{k} & 1 & & \\ & \lambda_{k} & 1 & & \\ & & \ddots & \ddots & \\ & & & \lambda_{k} & 1 \\ & & & & \lambda_{k} \end{bmatrix},$$
(3.11)

and $J \in \mathbb{R}^{2n_k}$ for $k = p + 1, \dots, q$ are the Jordan blocks for conjugate eigenvalues $\alpha_k \pm j\beta_k, \alpha_k > 0$ and $\beta_k > 0$, with the multiplicity n_k in the form,

$$J_{k} = \begin{bmatrix} \mu(\alpha_{k}, \beta_{k}) & I_{2} & & \\ & \mu(\alpha_{k}, \beta_{k}) & I_{2} & & \\ & & \ddots & \ddots & \\ & & & \mu(\alpha_{k}, \beta_{k}) & I_{2} \\ & & & & \mu(\alpha_{k}, \beta_{k}) \end{bmatrix},$$
(3.12)

with I_2 the identity matrix $\mathbb{R}^{2 \times 2}$ and,

$$\mu(\alpha_k, \beta_k) = \begin{bmatrix} \alpha_k & \beta_k \\ \beta_k & \alpha_k \end{bmatrix} \in \mathbb{R}^{2 \times 2}.$$

3.2.3. Jordan Matrix Decomposition Example

Consider the non-singular matrices $U \in \mathbb{R}^{n \times n}$ and $U^{-1} \in \mathbb{R}^{n \times n}$, such that,

$$U^{-1}\mathcal{L}U = J, \tag{3.13}$$

with J being a block-diagonal matrix of real Jordan form. To illustrate, consider a group of four agents with the topology shown in Figure 3.3 (Ding, 2014).



Figure 3.3 Topology for the Jordan decomposition example

The corresponding Laplacian matrix is,

$$\mathcal{L}_{1} = \begin{bmatrix} 1 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & -1 & 0 & 1 \end{bmatrix}.$$
 (3.14)

The eigenvalues of \mathcal{L}_1 are $\left[0, 1, \left(3 + \sqrt{3}j\right)/2, \left(3 - \sqrt{3}j\right)/2\right]$. Then,

$$J_{1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{3}{2} & \frac{\sqrt{3}}{2} \\ 0 & 0 & -\frac{\sqrt{3}}{2} & \frac{3}{2} \end{bmatrix}.$$
 (3.15)

Now, one can find U that satisfies the transformation as,

$$U = \begin{bmatrix} 1 & 0 & \frac{1}{2} & \frac{\sqrt{3}}{2} \\ 1 & 0 & -1 & 0 \\ 1 & -2 & \frac{1}{2} & \frac{\sqrt{3}}{2} \\ 1 & 0 & \frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix},$$
(3.16)

and,

$$U^{-1} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ \frac{1}{2} & 0 & -\frac{1}{2} & 0 \\ \frac{1}{3} & -\frac{2}{3} & 0 & \frac{1}{3} \\ \frac{\sqrt{3}}{3} & 0 & 0 & -\frac{\sqrt{3}}{3} \end{bmatrix}.$$
 (3.17)

3.2.4. Kronecker Product

In consensus control, the Kronecker product is used to describe the connections in multi-agent systems with respect to the Laplacian matrix structure. For $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{p \times q}$, the Kronecker product of A and B, denoted $A \otimes B$, is a $mp \times nq$ matrix defined as,

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{bmatrix},$$
 (3.18)

that satisfies,

$$A \otimes (B \otimes C) = (A \otimes B) \otimes C \tag{3.19}$$

$$(A \otimes B)(C \otimes D) = (AC) \otimes (BD) \tag{3.20}$$

$$(A \otimes B)^T = A^T \otimes B^T. \tag{3.21}$$

3.3. Lyapunov Stability

In order to perform stability analysis of nonlinear systems, the applied Lyapunov's second method is presented. This direct method allows to examine the stability of a dynamic system without explicitly solving the non-linear differential equation corresponding to the dynamics of the system.

Definition 3.3. The equilibrium point $x^* = 0$ of a system starting at an initial condition $x(t_0) = x_0$ is said to be Lyapunov stable if, for any R > 0, there exists some r(R) > 0 such that,

$$||x_0|| < r \Rightarrow ||x|| < R, \quad \forall t \ge t_0 . \tag{3.22}$$

Moreover, $x^* = 0$ is said to be asymptotically stable if there exists some r > 0 such that,

$$||x_0|| < r \Rightarrow \lim_{t \to \infty} ||x|| = 0.$$
(3.23)

Otherwise, the equilibrium point is unstable. This stability concept is illustrated in Figure 3.4 (Nguyen, 2018).

Definition 3.4. A function V(x) is a Lyapunov function if V(x) > 0 and $\dot{V} < 0$.

Letting Δ denote a domain around the equilibrium x = 0, consider a nonlinear system,

$$\dot{x} = f(x), \tag{3.24}$$

where $x \in \Delta \subset \mathbb{R}^n$ is the state of the system, and $f : \Delta \subset \mathbb{R}^n \to \mathbb{R}^n$ is a continuous function with x = 0 as an equilibrium point, that is f(0) = 0, and with x = 0 as an interior point of Δ . The following definition and theorem ensue.

Definition 3.5 (Radially Unbounded Functions). A C^1 function $V(x) \in \mathbb{R}^+$ with V(0) = 0 is said to be a radially unbounded function if $||x|| \to \infty \Rightarrow V(x) \to \infty$.



Figure 3.4 Stability concept. A system with initial conditions close to the origin is Lyapunov stable if the trajectory of the system can be kept arbitrarily close to it (Nguyen, 2018).

Theorem 3.3 (LaSalle's Invariance Principle). For the system defined by

Equation 3.24, if there exists a radially unbounded Lyapunov function $V(x) : \mathbb{R}^n \to \mathbb{R}$, then the equilibrium point $x^* = 0$ is globally asymptotically stable (Khalil, 2002).

As an example, consider the problem of stabilizing the linear system,

$$\dot{x} = Ax + Bu, \tag{3.25}$$

where $x \in \mathbb{R}^n, A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{n \times m}, u \in \mathbb{R}^m$. It is sufficient to choose the control law,

$$u = Kx, (3.26)$$

where $K \in \mathbb{R}^{m \times n}$. Consider a Lyapunov function candidate for the closed-loop dynamics as,

$$V(x) = \frac{1}{2}x^{T}Px,$$
 (3.27)

where P > 0 is a symmetric matrix. The time derivative of V(x) is given as,

$$\dot{V}(x) = x^T \left[(A + BK)^T P + P(A + BK) \right] x.$$
 (3.28)

The Lyapunov function candidate Equation 3.27 is a real Lyapunov function, such that the closed-loop dynamics are asymptotically stable if K is the solution of,

$$(A + BK)^T P + P(A + BK) = -Q, (3.29)$$

where Q is an arbitrary positive definite matrix.

3.4. Consensus Protocols in Networked Multi-Agent Systems

To analyze the stability of multi-agent systems, the highly complex dynamics of the individual agents are often simplified or neglected. The focus is shifted towards the study of their interactions and, with that regard, only information transfers between agents are considered. In this section, first-order (single integrator), second-order (double integrator) and higher-order multi-agent systems are presented.

The consensus problem, which is the development of a consensus algorithm for each agent based on its local information such that the group of agents can reach an agreement on certain quantities of interest, is a basic MAS coordination problem. The key idea is to drive the information variable of each agent towards the information variable of its neighbors (Ren & Beard, 2008).

3.4.1. First and Second-Order Consensus Algorithms

Consider a group of n agents. The information states with single-integrator dynamics are given by,

$$\dot{x}_i = u_i, \quad i = 1, \dots, n,$$
 (3.30)

where $x_i \in \mathbb{R}^m$ is the information state and $u_i \in \mathbb{R}^m$ is the control input of agent *i*.

Definition 3.6 (Distributed Control Protocols). The control given by,

$$u_i = k_i(x_{i_1}, \ x_{i_2}, \dots, x_{i_m}), \tag{3.31}$$

for some function $k_i(.)$ is said to be distributed if $m < n, \forall i$, that is, the control input of each node depends on some proper subset of all the nodes. It is said to be a protocol with topology \mathcal{G} if $u_i = k_i(x_i, \{x_j | j \in \mathcal{N}_i\})$, that is, each node can obtain information about the state only of itself and its in-neighbors in \mathcal{N}_i .

Definition 3.7 (Consensus Problem). Find a distributed control protocol that drives all states to the same values $x_i = x_j$, as $t \to \infty$, $\forall i, j$. This value is known as a consensus value (Lewis et al., 2014).

The fundamental consensus algorithm is given by,

$$u_i = -\sum_{j=1}^n a_{ij}(x_i - x_j), \qquad (3.32)$$

where a_{ij} is the $(i, j)^{th}$ entry of the adjacency matrix associated with the communication graph. Consensus is said to be achieved if $|x_i(t) - x_j(t)| \to 0$ as $t \to \infty$. This control is distributed in that it depends only on its immediate neighbors \mathcal{N}_i . In addition, this algorithm can be modified accordingly to achieve different control objectives, such as rendezvous (also called consensus), axial alignment and formation maneuvering (Beard et al., 2006; Fax & Murray, 2004; Lin et al., 2004).

The goal is to show that the protocol in Equation 3.32 solves the consensus problem. Writing the closed-loop dynamics as,

$$\dot{x}_i = -\sum_{j=1}^n a_{ij}(x_i - x_j), \qquad (3.33)$$

the global dynamics are given by,

$$\dot{x} = -\mathcal{D}x + \mathcal{A}x = -\mathcal{L}x. \tag{3.34}$$

For double-integrator dynamics, the states are,

$$\dot{x}_i = v_i, \quad \dot{v}_i = u_i, \quad i = 1, \dots, n,$$
(3.35)

where $v_i \in \mathbb{R}^m$ is the information state derivative. The general second-order consensus algorithm is,

$$u_{i} = -\sum_{j=1}^{n} a_{ij} \left[(x_{i} - x_{j}) + \gamma (v_{i} - v_{j}) \right], \qquad (3.36)$$

where $\gamma > 0$ is a control gain.

3.4.2. High-Order Consensus Algorithms

The goal of a consensus algorithm is to derive a control law u_i such that a consensus is reached among agents. A more general formulation of Equation 3.32 can be obtained by considering information variables with l^{th} -order dynamics given by

$$\dot{x_i}^{(0)} = x_i^{(1)}$$

 \vdots
 $\dot{x_i}^{(l-2)} = x_i^{(l-1)}$
 $\dot{x_i}^{(l-1)} = u_i,$

where $x_i^{(k)} \in \mathbb{R}^m$, k = 0, 1, ..., l - 1, are the states, $u_i \in \mathbb{R}^m$ is the control input, and $x_i^{(k)}$ denotes the k^{th} derivative of x_i with $x_i^{(0)} = x_i$, i = 1, ..., n. The following higher-order consensus algorithm is proposed in Ren et al. (2007),

$$u_{i} = -\sum_{j \in \mathcal{N}_{i}} a_{ij} k_{ij} \left[\sum_{k=0}^{l-1} \gamma_{k} \left(x_{i}^{(k)} - x_{j}^{(k)} \right) \right], \quad i = 1, \dots, n,$$
(3.37)

where $k_{ij} > 0$ and $\gamma_k > 0$ are the control gains. The motivation behind Equation 3.37 is to drive each vehicle's information variable and its high-order derivatives toward the states of its neighbors. Note that the linear consensus strategies reported in the literature can be considered special cases of Equation 3.37 when l = 1 or l = 2.

3.4.3. Consensus with Reference Tracking

Consensus algorithms for first-order dynamics to a reference model are presented in this section. Assume a team of n vehicles with an additional (virtual) agent labeled n + 1, where agent n + 1 is the team leader and vehicles $1, \ldots, n$ are the followers. The leader has access to the consensus reference state x^r , which satisfies,

$$\dot{x}^r = f(t, x^r), \tag{3.38}$$

where $f(\cdot, \cdot)$ is bounded, piecewise continuous in t and locally Lipschitz in x^r (Ren & Beard, 2008). For states with first-order dynamics, a consensus tracking algorithm with a constant reference state is given by,

$$u_i = -\sum_{j=1}^n a_{ij}(x_i - x_j) - a_{i(n+1)}(x_i - x^r), \quad i = 1, \dots, n,$$
(3.39)

where $x_i \in \mathbb{R}^m$ is the i^{th} state and a_{ij} is the (i, j) entry of the adjacency matrix $A_{n+1} \in \mathbb{R}^{(n+1) \times (n+1)}$.

3.5. Quadrotor Dynamics

Quadrotors are versatile aircraft with two main characteristics, namely under-actuation and coupling in roll-pitch-yaw. To design an adequate station-keeping and tracking consensus controller, a dynamics inversion technique is used in this research.



Figure 3.5 Quadrotor forces and moments (Stirling et al., 2012)

Assume a rigid body vehicle with six degrees of freedom with relative position in the Earth reference frame defined by $[x, y, z]^T$, and the three Euler angles representing the attitude by $[\phi, \theta, \psi]^T$ being the roll, pitch and yaw, respectively. The rotations are represented by the transformation matrix R defined as,

$$R = \begin{pmatrix} c_{\theta}c_{\psi} & s_{\phi}s_{\theta}c_{\psi} - c_{\phi}s_{\psi} & c_{\phi}s_{\theta}c_{\psi} + s_{\phi}s_{\psi} \\ c_{\theta}s_{\psi} & s_{\phi}s_{\theta}s_{\psi} + c_{\phi}c_{\psi} & c_{\phi}s_{\theta}s_{\psi} - s_{\phi}c_{\psi} \\ -s_{\theta} & s_{\theta}c_{\phi} & c_{\phi}c_{\theta} \end{pmatrix} , \qquad (3.40)$$

where c_{ϕ} and s_{ϕ} represent $\cos \phi$ and $\sin \phi$, respectively. This matrix transforms vectors from the body frame, where the forces are defined, to the inertial frame. It is obtained by a succession of rotations around the roll, pitch and yaw angles (Altug, Ostrowski, & Taylor, 2005). Each rotor produces moments as well as vertical forces. These moments have been experimentally observed to be linearly dependent on the forces for low speeds. The quadrotor is an under-actuated system with four input forces (F_1 , F_2 , F_3 , F_4) and six output states (x, y, z, ϕ , θ , ψ). The equations of motion can be written using the forces F_{1-4} and moments M_{1-4} ,

$$\begin{cases} \ddot{x} = \left[\left(\sum_{i=1}^{4} F_{i} \right) \left(\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \right) - K_{1} \dot{x} \right] / m \\ \ddot{y} = \left[\left(\sum_{i=1}^{4} F_{i} \right) \left(\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \right) - K_{2} \dot{y} \right] / m \\ \ddot{z} = \left[\left(\sum_{i=1}^{4} F_{i} \right) \left(\cos \phi \cos \theta \right) - mg - K_{3} \dot{z} \right] / m \\ \ddot{\phi} = l (-F_{1} + F_{2} + F_{3} - F_{4} - K_{4} \dot{\phi}) / I_{1} \\ \ddot{\theta} = l (-F_{1} - F_{2} + F_{3} + F_{4} - K_{5} \dot{\theta}) / I_{2} \\ \ddot{\psi} = (M_{1} - M_{2} + M_{3} - M_{4} - K_{6} \dot{\psi}) / I_{3} , \end{cases}$$
(3.41)

where m is the quadrotor mass, g is the acceleration of gravity and l is the length from the center of mass to the rotor. K_{1-6} represent the drag coefficients and I_{1-3} are the moments of inertia with respect to the axes.

Since drag is negligible at low speeds, it is assumed to be zero. The inputs are

defined using the following control allocation,

$$u_1 = (F_1 + F_2 + F_3 + F_4)/m \tag{3.42}$$

$$u_2 = (-F_1 + F_2 + F_3 - F_4)/I_1$$
(3.43)

$$u_3 = (-F_1 - F_2 + F_3 + F_4)/I_2 \tag{3.44}$$

$$u_4 = C(F_1 - F_2 + F_3 - F_4)/I_3, (3.45)$$

where C is the force-to-moment scaling factor (Altug et al., 2005).

The equations of motions from Equation 3.41 become,

$$\begin{cases} \ddot{x} = u_1(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) \\ \ddot{y} = u_1(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi) \\ \ddot{z} = u_1(\cos\phi\cos\theta) - g \\ \ddot{\phi} = u_2l \\ \ddot{\theta} = u_3l \\ \ddot{\psi} = u_4, \end{cases}$$
(3.46)

where u_1 is the normalized total lift force, and u_2 , u_3 and u_4 correspond to the control inputs of roll, pitch and yaw moments, respectively. They can be considered as inputs to the system due to the linear relationship with the thrust forces generated by the four rotors. Equation 3.46 represents an under-actuated nonlinear coupled system with six outputs and four inputs (Das et al., 2009; Altug et al., 2005).

3.6. Feedback Linearization for Quadrotor Dynamics

Following the same procedure presented in Wang et al. (2013), the nonlinear coupled system in Equation 3.46 can be transformed into an equivalent linearized system such that classical control methods can be applied. This approach transforms the high-order nonlinear coupled system into a term of four linear subsystems so that consensus algorithms can be designed based on these equivalent subsystems.

Since the dynamic Equation 3.46 is an under-actuated system, feedback linearization cannot be performed directly. To overcome this, four outputs $y_1 = [z, \phi, \theta, \psi]^T$ are considered to control the altitude and attitude of the quadrotor. However, this introduces zero dynamics effects, which result in the drift of the quadrotor in the x - y plane. Acceleration along the x-axis or y-axis will occur even for small tilt angles, and the way to control these accelerations is to tilt the system in the opposite direction. To suppress the effects of these undesired dynamics, another controller or a combination of controllers is required.

Feedback linearization is successful if the zero dynamics are stabilized. First, the following assumptions need to be made,

- 1. Under nominal flight conditions, the quadrotor does not perform large angle maneuvers. The pitch angle and roll angle of every quadrotor is therefore assumed to be bounded in the range $\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$.
- 2. The equilibrium of the system in Equation 3.46 is at this point satisfying $\phi = 0, \ \theta = 0$ and $\psi = 0$.

Using these assumptions and setting $\ddot{z} = 0$, the zero dynamics of the system can

be written as $\ddot{x} = \tan \theta$ and $\ddot{y} = -\tan \phi / \cos \theta$. An approximate nonlinear state equation of the system in Equation 3.46 is then obtained as,

$$\begin{cases} \ddot{x} = g \tan \theta \\ \ddot{y} = -g \tan \phi \\ \ddot{z} = u_1 \cos \phi \cos \theta - g \\ \ddot{\phi} = u_2 l \\ \ddot{\theta} = u_3 l \\ \ddot{\psi} = u_4 . \end{cases}$$
(3.47)

Equation 3.47 shows that x is only related with θ , and decoupled with the input u_1 and states ϕ and ψ . Similarly, y is only related with ϕ , and decoupled with u_1 , θ and ψ . Equation 3.47 can then be split into four equivalent subsystems,

$$\begin{cases} \ddot{x} = g \tan \theta \\ \ddot{\theta} = u_3 l \end{cases}$$
(3.48a)

$$\begin{cases} \ddot{y} = -g \tan \phi \\ \ddot{\phi} = u_2 l \end{cases}$$
(3.48b)

$$\ddot{z} = u_1 \cos\phi \cos\theta - g \tag{3.48c}$$

$$\ddot{\psi} = u_4 . \tag{3.48d}$$

$$\begin{cases} \ddot{x} = v \\ \ddot{\theta} = u_3 l \end{cases}$$
(3.49a)

$$\begin{cases} \ddot{y} = w \\ \ddot{\phi} = u_2 l \end{cases}$$
(3.49b)

$$\ddot{z} = u_1 \cos\phi \cos\theta - g \tag{3.49c}$$

$$\ddot{\psi} = u_4 . \tag{3.49d}$$

To linearize this system of equations and obtain the inputs, the equations are differentiated until the inputs appear. A fourth-order derivative of the states is therefore necessary. Using the x-dynamics as an example, the following derivation is performed,

$$v = \ddot{x} = g \tan \theta \tag{3.50a}$$

$$\dot{v} = g(1 + \tan^2 \theta)\dot{\theta} \tag{3.50b}$$

$$\ddot{v} = u_3 l \left(g + \frac{v^2}{g} \right) + \frac{2v\dot{v}^2}{g^2 + v^2} .$$
(3.50c)

The dynamics in Equation 3.49 are now transformed into four decoupled subsystems,

$$\begin{cases} \ddot{x} = v \\ \ddot{v} = u'_3 \end{cases}$$
(3.51a)

$$\begin{cases} \ddot{y} = w \\ \ddot{w} = u'_2 \end{cases}$$
(3.51b)

$$\ddot{z} = u_1' \tag{3.51c}$$

$$\ddot{\psi} = u_4,$$
 (3.51d)

where the virtual inputs u_1' , u_2' and u_3' are defined as,

$$u_1' = u_1 \cos\phi\cos\theta - g \tag{3.52}$$

$$u_{2}' = -u_{2}l\left(g + \frac{w^{2}}{g}\right) + \frac{2w\dot{w}^{2}}{g^{2} + w^{2}}$$
(3.53)

$$u'_{3} = u_{3}l\left(g + \frac{v^{2}}{g}\right) + \frac{2v\dot{v}^{2}}{g^{2} + v^{2}}.$$
(3.54)

Finally, the new control inputs u_{1i} , u_{2i} , u_{3i} and u_{4i} are extracted from the virtual control inputs and are given as,

$$u_{1i} = \frac{u'_{1i} + g}{\cos \phi_i \cos \theta_i} \tag{3.55}$$

$$u_{2i} = \frac{u'_{2i} - 2w\dot{w}^2/(g^2 + w^2)}{-l\left(g + w^2/g\right)}$$
(3.56)

$$u_{3i} = \frac{u'_{3i} - 2v\dot{v}^2/(g^2 + v^2)}{l\left(g + v^2/g\right)}$$
(3.57)

$$u_{4i} = -\beta_1 \psi - \beta_2 \dot{\psi}, \qquad (3.58)$$

where β_1 and β_2 are gains for the yaw. The control inputs u'_{1i} , u'_{2i} and u'_{3i} will later be augmented by the adaptive mechanism proposed in this research to increase robustness in response to potential upset conditions in the agents. The design process follows hereafter in Chapter 4.

Summary: This chapter presented the mathematical concepts applied in this dissertation and explained the fundamentals for the study. A review of graph theory, matrix theory and Lyapunov stability was presented. Consensus protocols and high-order consensus algorithm were then examined followed by the linear and nonlinear equations that are required to model high-order systems.

4. Immunity-Inspired Distributed MRAC

The immunity-inspired architecture for global disturbance rejection is presented in this chapter along with details of the distributed consensus architecture. Both the local and global self are also defined. As introduced in Chapter 1, a successful algorithm for a cooperative task operation of different autonomous agents has the advantages of reducing costs, enhancing configurability, robustness, and it provides potential in its application in scientific and military purposes. Current definitions of autonomy based on self properties rely mostly on a rather rigid initial framework with low capabilities of modifications in response to the environment. Designing an efficient cooperative operation scheme is still a challenge due to the complexity and nonlinearities of the overall system induced by disturbances typical of highly uncertain and unpredictable environments.

4.1. Intelligent Systems

In the past years, intelligent systems have attracted much attention in the artificial intelligence community as a solution to cooperative mission problems, and several methods have since been proposed. However, earlier developments of MAS coordination formulations were based on a centralized operation which potentially might put the mission objectives at risk in case of limited communication, failure of the leader agent to perform specific tasks, or unavailability of human operators (F. & Moncayo, 2021).

The immune system is a highly intelligent cooperative system with trillions of antibodies that can self-organize to produce a proper and adequate response to invading antigens. Such biological systems exhibit robust, adaptive, and highly distributed cognitive capabilities. Agents within the immune system work together to afford a level of protection for the host to maintain a steady operation state. The biological immune system provides a natural protection against pathogens by identifying and destroying harmful cells or molecules. This functioning is similar to how a robust adaptive system is equipped to deal with a variety of disturbances and uncertainties. Furthermore the immune system memorizes how previous encounters were successfully defeated and responds faster to similar encounters in the future. This is due to the distinction between *self* and *non-self*, which respectively represent the known nominal functioning of the host and the anomalous behavior. Non-self refers to any system behavior that does not agree with the self. These characteristics provide the general conceptual basis for developing an integrated and comprehensive framework to enhance MAS mission performance.

One of the main goals herein is to design an adaptive tracking controller for a resilient MAS by using a distributed approach. Consensus is then achieved from relative information, namely position and velocity. The objective is for the agents to achieve consensus and maintain their positions and orientations when each agent is provided with information only from neighboring agents in the directed graph.

4.2. Immunity-Inspired Architecture

The derived immune mechanism model can be applied within a more general multi-agent architecture, such as the one described in F. & Moncayo (2020). This architecture is depicted in Figure 4.1 and represents the interaction of an agent (*local self*) within its multi-agent network (*global self*) to increase resilience and autonomy of a cooperative multi-agent mission. The work presented in this chapter focuses on

providing a solution for the local self component. The global self component is detailed in Chapter 7.

4.2.1. Local Self

Within the local self, each agent features initial immune characteristics to reject local faults, failures and disturbances. During a typical mission operation, an agent is vulnerable and subject to threats. For this local self, a threat may refer to hardware failures (actuators, sensors, onboard computer, power source) and operational abnormal conditions or exogenous disturbances (weather, GPS denied location, cyber-attacks). Even though these threats are unpredictable, their effects on the system and the multi-agent network can be abated if assessed and processed in time. This is possible with the presence of a health management (HM) strategy within each individual agent. Once an agent detects a threat, a signal can be generated to the network. This signal is processed within a global interaction topology, where neighboring agents would notice the change and will trigger collaborative actions to maintain mission performance.

The proposed approach for the local HM framework relies on the AIS paradigm. This method depends on the self/nonself definition of the system using extensive data collection of different features during nominal and upset conditions. With this first self/nonself representation, each agent will have initial fault rejection capabilities. Through the implementation of an online HM scheme, each agent will then develop its own immune capabilities to monitor and reject local disturbances. In time, each local self will learn more about its environment, building and developing its own robust immune system.





4.2.2. Global Self

As the immune system is set up to detect and eliminate invasive antigens, so is the aim of MAS coordination in flight to eliminate threats. When referring to the global self, a threat could represent any unexpected event that may risk the MAS mission such as physical obstacles, faulty agents, extreme environmental conditions, radar areas or communication cyber-attacks. Each agent within the MAS has information of both local and global self conditions, such that any abnormal behavior of any agent is sensed by the whole network. The manner in which the global self uses this information is presented and illustrated in Chapter 7.

4.3. Immunity-Inspired Adaptation Law

The AIS paradigm is inspired on different immune mechanisms typical in living organisms. It is made up of the interaction of specialized cells that target and rid the body of infectious cells. The mathematical representation of the AIS, originally shown in Takahashi & Yamada (1998), is presented in this section.

Agents within the immune system work together to afford a level of protection to the host that maintains a steady operation state. T and B cells are types of white blood cells that determine the specificity of immune responses to foreign substances referred to as antigens. The body's adaptive immune response provides the vertebrate immune system with the ability to recognize and remember specific pathogens to generate immunity and mount stronger attacks each time the pathogen is encountered. A simplified representation of this ongoing process, known as the humoral feedback mechanism, is represented in Figure 4.2. The antigen-specific response of T and B cells includes antibody production and the eradication of pathogen-infected cells. It is regulated by cytokines which are regulatory proteins released by cells of the immune system and act as inter-cellular mediators in the generation of an immune response. The immune cells are able to learn and improve immune defenses when they encounter the same pathogen several times. This is based on the concept of "memory" in certain immune cells such as the T and B cells.



Figure 4.2 Humoral feedback mechanism

Although immune system mechanisms are complex and difficult to characterize, a simple model that represents the interaction between T and B cells at a given instant time k can be derived following the difference between the amount of T-helper



Figure 4.3 T_s -cells nonlinear function (A. E. Rocha, 2016)

cells (T_h) and T-suppressing cells (T_s) ,

$$B(k) = T_h(k) - T_s(k) . (4.1)$$

If the total amount of antigens at time k is defined as $\lambda(k)$, then the response of the T_h -cells can be represented as,

$$T_h(k) = c_1 \lambda(k), \tag{4.2}$$

where c_1 is a stimulation constant (W. Chen et al., 2006). Similarly, the production of T_s -cells can be defined as,

$$T_s(k) = c_2 f(\Delta B(k))\lambda(k), \qquad (4.3)$$

where c_2 is a suppression constant, $\Delta B(k)$ is the change of concentration of *B*-cells, and $f(\Delta B(k))$ is a nonlinear function that relates this change with the amount of T_s -cells, as in the example shown in Figure 4.3. In combining these equations, one can obtain a more general equation that represents these immune interactions,

$$B(k) = K[1 - \eta f(\Delta B(k))]\lambda(k), \qquad (4.4)$$

where K is a reaction rate and $\eta = c_2/c_1$ is a factor that characterizes the dynamic interaction between the T_h -cells and T_s -cells. In general, the stability will depend on the parameter η and the nonlinear function $f(\Delta B(k))$. It is noticeable that Equation 4.4 represents an adaptive feedback mechanism where the amount of *B*-cells (input u(k)) regulates and minimizes the total amount of antigens (error e(k)) present in the system. Therefore, from a controls perspective, Equation 4.4 becomes,

$$u(k) = K [1 - \eta f(\Delta u(k))] e(k) .$$
(4.5)

4.4. Distributed Consensus Architecture

In this research, a local adaptation law is proposed as part of a consensus controller previously presented in Wang et al. (2013). A baseline controller is first designed for each agent following the feedback linearization strategy described in Section 3.6 as a first layer in the control architecture. Since exact feedback linearization is not usually achieved, especially in real applications, a second layer includes an adaptive augmentation designed to increase robustness by eliminating residual nonlinearities that might still be present in the system. This augmentation relies on a model-reference architecture that is inspired by the immune response mechanism. The advantage of using this bio-inspired approach is the introduction of a distributed self-adaptive system that allows fast response to hostile invasions (e.g.
disturbances or failures).

4.4.1. Quadrotor Dynamics

In Chapter 3, the linearized decoupled model of the quadrotor given in

Equation 3.51 was obtained through dynamic inversion. The quadrotor dynamics for agent i can be rewritten in a matrix form as,

$$\dot{\mathbf{x}}_i = A\mathbf{x}_i + B(u_i' + \delta_i) , \qquad (4.6)$$

where
$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$
, $B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$, and δ is a bounded, time-varying disturbance.

The vector \mathbf{x}_i is the state vector representing either $[x_i \ \dot{x}_i \ v_i \ \dot{v}_i]^T$ in the *x*-direction or $[y_i \ \dot{y}_i \ w_i \ \dot{w}_i]^T$ in the *y*-direction. In the *z*-direction, $\mathbf{x}_i = [z_i \ \dot{z}_i]^T$ and the *A* and *B* matrices become $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

4.4.2. Reference Model

A reference model is used to specify a desired response of an adaptive control system to a command input. In an adaptive controller, the adaptation is operated on the tracking error between the reference model and the system output (Nguyen, 2018). By choosing the structure and parameters of a reference model suitably, its outputs can be used as the desired plant response. The proposed adaptive controller then follows this reference model after the disturbance is injected in the MAS.

Assume a nominal reference model for each agent i with desired dynamics,

$$\dot{\mathbf{x}}_{m_i} = A\mathbf{x}_{m_i} + Bu_{m_i} , \qquad (4.7)$$

where $\mathbf{x}_{m_i} = [x_{m_i}, \dot{x}_{m_i}, v_{m_i}, \dot{v}_{m_i}]$ is the reference model state vector. The reference control law u_{m_i} is given by,

$$u_{m_i} = K_1 \mathbf{x}_{m_i} - K_2 \sum_{j=1}^n a_{ij} (\mathbf{x}_{m_i} - \mathbf{x}_{m_j}) , \qquad (4.8)$$

where K_1 is a local feedback gain that is equal to,

$$K_{1} = \begin{cases} -k_{\dot{z}} & \text{for the } z\text{-direction states} \\ -[k_{\dot{y}} \quad k_{w} \quad k_{\dot{w}}] & \text{for the lateral states} \\ -[k_{\dot{x}} \quad k_{v} \quad k_{\dot{v}}] & \text{for the longitudinal states} , \end{cases}$$
(4.9)

and K_2 is a consensus gain to be defined later. Equation 4.8 can be expressed as follows for all states,

$$\ddot{z}_{m_i} = -k_{\dot{z}}\dot{z}_{m_i} - K_2 \sum_{j=1}^n a_{ij}(z_{m_i} - z_{m_j})$$
(4.10a)

$$\ddot{y}_{m_{i}} = -k_{\dot{y}}\dot{y}_{m_{i}} - k_{w}w_{m_{i}} - k_{\dot{w}}\dot{w}_{m_{i}} - K_{2}\sum_{j=1}^{n}a_{ij}\left(\begin{bmatrix}y_{m_{i}}\\\dot{y}_{m_{i}}\\w_{m_{i}}\end{bmatrix} - \begin{bmatrix}y_{m_{j}}\\\dot{y}_{m_{j}}\\w_{m_{j}}\end{bmatrix}\right)$$
(4.10b)
$$n\left(\begin{bmatrix}x_{m_{i}}\end{bmatrix} - \begin{bmatrix}x_{m_{j}}\end{bmatrix}\right)$$

$$\ddot{x}_{m_{i}} = -k_{\dot{x}}\dot{x}_{m_{i}} - k_{v}v_{m_{i}} - k_{\dot{v}}\dot{v}_{m_{i}} - K_{2}\sum_{j=1}^{n}a_{ij}\left(\begin{bmatrix} x_{m_{i}}\\ \dot{x}_{m_{i}}\\ v_{m_{i}}\end{bmatrix} - \begin{bmatrix} x_{m_{j}}\\ \dot{x}_{m_{j}}\\ v_{m_{j}}\end{bmatrix}\right).$$
 (4.10c)

This model allows for reference trajectories to be generated for each agent, using available information from its neighbors. It follows that the agents can have different reference models, depending on the current topology. A set of control laws can now be derived such that the reference model is properly tracked with bounded adaptation and guaranteeing stability of the tracking error dynamics of the closed loop system (A. E. Rocha, 2016). Accordingly, this reference model outputs an ideal trajectory which is used to generate the tracking error e_i , for $i = 1, \ldots, n$, given by,

$$e_i = \mathbf{x}_i - \mathbf{x}_{m_i} \ . \tag{4.11}$$

The objective of the MRAC is to keep the tracking error e_i as small as possible by adapting to the system uncertainty, that is, $e_i(t) \to 0$. In this ideal case, the system state follows the model-reference signal perfectly, that is $\mathbf{x}_i(t) \to \mathbf{x}_{m_i}(t)$, and according to Definition 3.7, consensus is achieved.

4.4.3. Baseline Control Law

In this dissertation, the first level of control for each agent is a distributed consensus protocol referred to as the *baseline controller*, u_{b_i} , and is defined as,

$$u_{b_i} = u_{l_i} + u_{c_i} \tag{4.12}$$

This controller will serve as a benchmark when testing the efficacy of the adaptive controller.

The first part, u_{l_i} , is the local feedback stability controller and is given as,

$$u_{l_i} = K_1 \mathbf{x}_i, \tag{4.13}$$

where K_1 (Equation 4.9) is a feedback gain matrix that can be designed using pole placement. This control law ensures stability of each quadrotor.

The second part of Equation 4.12, u_{c_i} , is a high-order consensus algorithm representing the interactions with the neighbors. It is given by,

$$u_{c_i} = -K_2 \sum_{j=1}^{n} a_{ij} (\mathbf{x}_i - \mathbf{x}_j), \qquad (4.14)$$

where K_2 is the consensus gain.

Finally, Equation 4.12 becomes,

$$u_{b_i} = K_1 \mathbf{x}_i - K_2 \sum_{j=1}^n a_{ij} (\mathbf{x}_i - \mathbf{x}_j).$$
(4.15)

4.4.4. Adaptive Control Law

To mitigate the effects of the disturbances, a local adaptive control augmentation is implemented. It is defined as,

$$u_{AD_i} = -K_{AD}(\mathbf{x}_i - \mathbf{x}_{m_i}). \tag{4.16}$$

A successful adaptive augmentation controller should have the ability to mitigate local uncertainties or agent subsystem malfunctions.

Following the aforementioned reference model and immunity-based feedback mechanism model from Equation 4.5, an adaptive law is designed and applied to the virtual gains by considering the implicit adaptivity of,

$$\Delta u_i(t) = u_{m_i}(t) - u_i(t) , \qquad (4.17)$$

where $u_{m_i}(t)$ is the nominal model-reference plant control input and $u_i(t)$ is the closed-loop non-adaptive control input. The local adaptive controller for each output can then be defined as,

$$u_{AD_{i}}(t) = \begin{bmatrix} u_{z_{AD_{i}}}(t) \\ u_{w_{AD_{i}}}(t) \\ u_{v_{AD_{i}}}(t) \\ u_{v_{AD_{i}}}(t) \\ u_{\psi_{AD_{i}}}(t) \end{bmatrix} = -\begin{bmatrix} k_{ez_{i}}(t)(z_{i} - z_{m_{i}}) + k_{e\dot{\omega}_{i}}(t)(\dot{z}_{i} - \dot{z}_{m_{i}}) \\ k_{ew_{i}}(t)(w_{i} - w_{m_{i}}) + k_{e\dot{\omega}_{i}}(t)(\dot{w}_{i} - \dot{w}_{m_{i}}) \\ k_{ev_{i}}(t)(v_{i} - v_{m_{i}}) + k_{e\dot{\omega}_{i}}(t)(\dot{v}_{i} - \dot{v}_{m_{i}}) \\ k_{e\psi_{i}}(t)(\psi_{i} - \psi_{m_{i}}) + k_{e\dot{\psi}_{i}}(t)(\dot{\psi}_{i} - \dot{\psi}_{m_{i}}) \end{bmatrix}$$

$$= - \begin{bmatrix} k_{ez_{i}}(t)e_{z_{i}} + k_{e\dot{z}_{i}}(t)\dot{e}_{z_{i}} \\ k_{ew_{i}}(t)e_{w_{i}} + k_{e\dot{w}_{i}}(t)\dot{e}_{w_{i}} \\ k_{ev_{i}}(t)e_{v_{i}} + k_{e\dot{v}_{i}}(t)\dot{e}_{v_{i}} \\ k_{e\psi_{i}}(t)e_{\psi_{i}} + k_{e\dot{\psi}_{i}}(t)\dot{e}_{\psi_{i}} \end{bmatrix}, \qquad (4.18)$$

where the adaptive gains are calculated by,

$$\begin{aligned} k_{ez_i}(t) &= k_{\dot{z}} k_z \eta f[\Delta u_{z_i}(t)] & k_{e\dot{z}_i}(t) = k_{\dot{z}} \eta f[\Delta u_{z_i}(t)] \\ k_{ew_i}(t) &= k_{\dot{w}} k_w \eta f[\Delta u_{w_i}(t)] & k_{e\dot{w}_i}(t) = k_{\dot{w}} \eta f[\Delta u_{w_i}(t)] \\ k_{ev_i}(t) &= k_{\dot{v}} k_v \eta f[\Delta u_{v_i}(t)] & k_{e\dot{v}_i}(t) = k_{\dot{v}} \eta f[\Delta u_{v_i}(t)] \\ k_{e\psi_i}(t) &= k_{\dot{\psi}} k_{\psi} \eta f[\Delta u_{\psi_i}(t)] & k_{e\dot{\psi}_i}(t) = k_{\dot{\psi}} \eta f[\Delta u_{\psi_i}(t)] . \end{aligned}$$

 $f[\Delta u(t)]$ represents the T_s -cell production. In this work, the following non-linear positive definite bounded function that incorporates adaptivity was selected,

$$f[\Delta u(t)] = 1 - \frac{-2\epsilon + 1}{e^{\gamma[\Delta u(t)]^2} + e^{-\gamma[\Delta u(t)]^2}}, \qquad (4.19)$$

where ϵ is the adaptive function bias (A. E. Rocha, 2016).

Considering the dynamics from Equation 3.51 and based on the general formulation for high-order consensus in Equation 3.37, the virtual controller for the i^{th} agent becomes,

$$u'_{1_{i}} = u_{1l_{i}} + u_{1AD_{i}} + u_{1c_{i}}$$

$$= \underbrace{-k_{\dot{z}}\dot{z}_{i}}_{u_{1l_{i}}} \underbrace{-k_{ez_{i}}(t)e_{z_{i}} - k_{e\dot{z}_{i}}(t)e_{\dot{z}_{i}}}_{u_{z_{AD_{i}}}(t)} \underbrace{-k\sum_{j=1}^{n} a_{ij}(z_{i} - z_{j})}_{u_{1c_{i}}}$$
(4.20)

$$u_{2_i}' = u_{2l_i} + u_{2AD_i} + u_{2c_i}$$

$$= \underbrace{-k_{\dot{y}}\dot{y}_{i} - k_{w}w_{i} - k_{\dot{w}}\dot{w}_{i}}_{u_{2l_{i}}} \underbrace{-k_{ew_{i}}(t)e_{w_{i}} - k_{e\dot{w}_{i}}(t)e_{\dot{w}_{i}}}_{u_{w_{AD_{i}}}(t)} - k\sum_{j=1}^{n}a_{ij}\left(\begin{bmatrix}y_{i}\\\dot{y}_{i}\\w_{i}\end{bmatrix} - \begin{bmatrix}y_{j}\\\dot{y}_{j}\\w_{j}\end{bmatrix}\right)}_{u_{2c_{i}}}$$
(4.21)

 $u_{3_i}' = u_{3l_i} + u_{3AD_i} + u_{3c_i}$

$$=\underbrace{-k_{\dot{x}}\dot{x_i}-k_vv_i-k_{\dot{v}}\dot{v_i}}_{u_{3l_i}}\underbrace{-k_{ev_i}(t)e_{v_i}-k_{e\dot{v}_i}(t)e_{\dot{v}_i}}_{u_{v_{AD_i}}(t)}$$

$$\underbrace{-k\sum_{j=1}^{n}a_{ij}\left(\begin{bmatrix}x_i\\\dot{x}_i\\v_i\end{bmatrix}-\begin{bmatrix}x_j\\\dot{x}_j\\v_j\end{bmatrix}\right)}_{u_{3c_i}}.$$
 (4.22)

4.4.5. Proposed DMRAC

A consensus protocol can be designed with the following structure,

$$u'_{i} = K_{1}\mathbf{x}_{i} - K_{2}\sum_{j=1}^{n} a_{ij}(\mathbf{x}_{i} - \mathbf{x}_{j}) - K_{AD}(\mathbf{x}_{i} - \mathbf{x}_{m_{i}}) .$$
(4.23)

Replacing Equation 4.23 within Equation 4.6, the closed-loop dynamics become,

$$\dot{\mathbf{x}}_i = A\mathbf{x}_i + BK_1\mathbf{x}_i - BK_2\sum_{j=1}^n a_{ij}(\mathbf{x}_i - \mathbf{x}_j) - BK_{AD}e_i + B\delta_i .$$
(4.24)

Figure 4.4 illustrates the general control architecture including the proposed adaptive augmentation.

4.5. Distributed MRAC for Formation Control

The tracking case for the MAS extends the proposed DMRAC to the formation control case. The goal is for the agents to follow a predefined trajectory while maintaining a set separation between them. The tracking consensus control law is given as,



Figure 4.4 Consensus control for a multi-agent system with immune adaptive augmentation (F. & Moncayo, 2021).

$$u_{c_{1i}} = -K_2 \sum_{j \in \mathcal{N}_i} a_{ij}(z_i - z_j)$$
(4.25)

$$u_{c_{2i}} = -K_2 \sum_{j \in \mathcal{N}_i} a_{ij} \left(\begin{bmatrix} y_i - \delta_{y_i} \\ \dot{y}_i \\ w_i \end{bmatrix} - \begin{bmatrix} y_j - \delta_{y_j} \\ \dot{y}_j \\ w_j \end{bmatrix} \right) - (y_i - \delta_{y_i} - y_r)$$
(4.26)

$$u_{c_{3i}} = -K_2 \sum_{j \in \mathcal{N}_i} a_{ij} \left(\begin{bmatrix} x_i - \delta_{x_i} \\ \dot{x}_i \\ v_i \end{bmatrix} - \begin{bmatrix} x_j - \delta_{x_j} \\ \dot{x}_j \\ v_j \end{bmatrix} \right) - (x_i - \delta_{x_i} - x_r) . \quad (4.27)$$

In this case, δ_{x_i} and δ_{y_i} represent the desired separations between the agents in the x and y axis, respectively. The type of formation control implemented is called distanced-based formation, where measurements contain only relative variables that can be sensed with respect to local coordinate systems of the agents (Oh & Ahn, 2011). Since the adaptive and feedback contributions are not changed by the new consensus law, only the consensus part of Equation 4.23 is modified.

Summary: The development of the distributed MRAC for agents within an immunity-inspired architecture was presented in this chapter, both for the consensus control case (regulator) and for the formation control case (tracking). The DMRAC endows each agent with disturbance-rejection capabilities. This architecture, furthered in Chapter 7, will provide the network with resilience from its global AIS.

5. Stability Analysis

The stability of the proposed adaptive consensus controller is analyzed in this chapter. Lyapunov's direct method, as shown in Section 3.3, is used to demonstrate that the controller achieves consensus asymptotically in the presence of bounded disturbances. For this purpose, the analyses are focused on the *x*-direction consensus protocol; the results can also be applied to the *y* and *z* directions. This work is the MAS adaptation of the proof presented in Rocha (2016). The following notations will be used throughout this chapter. $\lambda_i(A)$ represents the *i*th eigenvalue of a matrix A. \otimes denotes the Kronecker product. A > 0 means that A is a positive definite matrix. 'sup' and 'inf' denote the supremum and infinimum, respectively.

In Equation 4.24, the closed-loop dynamics are defined as,

$$\dot{\mathbf{x}}_i = A\mathbf{x}_i + BK_1\mathbf{x}_i - BK_2\sum_{j=1}^n a_{ij}(\mathbf{x}_i - \mathbf{x}_j) - BK_{AD}e_i + B\delta_i \ .$$

Defining the tracking error as $e_i = \mathbf{x}_i - \mathbf{x}_{m_i}$, the error dynamics can then be written as,

$$\dot{e}_i = Ae_i + BK_1e_i - BK_2\sum_{j=1}^n a_{ij}(e_i - e_j) - BK_{AD}e_i + B\delta_i .$$
(5.1)

In a more compact form, Equation 5.1 can be re-written as,

$$\dot{e} = \left[I_N \otimes \bar{A} + \mathcal{L} \otimes BK\right] e + \left(I_N \otimes B\right) \left[-\Psi(y_{\mathbf{x}}) + \delta\right] , \qquad (5.2)$$

where $\bar{A} = A + BK_1$, $K = -K_2$, $e = [e_1^T, \dots, e_N^T]^T$ and $\delta = [\delta_1^T, \dots, \delta_N^T]^T$.

The nonlinear adaptation term $\Psi(y_{\mathbf{x}})$ is defined as,

$$\Psi(y_{\mathbf{x}}) = \psi(t)y_{\mathbf{x}}$$
$$= \eta f(\Delta u) [k_D k e + k_D \dot{e}]$$
$$= \eta f(\Delta u)y, \qquad (5.3)$$

where $\psi(t) = \eta f(\Delta u)$ and $y = \overline{C}e$, with $\overline{C} = I_N \otimes C = I_N \otimes [k_D K, k_D]$.

Theorem 5.1. Consider the multi-agent system in Equation 4.24 with the nominal reference model given in Equation 4.7. Select the control law proposed in Equation 4.12 with $K = -B^T P$ and choose the adaptive control gains. Then, all trajectories in the closed-loop system Equation 4.24 are uniformly ultimately bounded, and,

$$\lim_{x\to\infty} \|\mathbf{x}_i - \mathbf{x}_{m_i}\| \le \gamma_1 \; ,$$

for some constant $\gamma_1 > 0$.

Proof. The following Lyapunov candidate function is proposed,

$$V = \frac{1}{2}e^{T}(\mathcal{L} \otimes P)e , \qquad (5.4)$$

where P > 0 is a solution to the following algebraic Riccati equation,

$$\bar{A}^T P + P\bar{A} - PBB^T P = -Q . (5.5)$$

The time derivative of Equation 5.4 gives,

$$\begin{split} \dot{V} &= \frac{1}{2} \dot{e}^{T} (\mathcal{L} \otimes P) e + \frac{1}{2} e^{T} (\mathcal{L} \otimes P) \dot{e} \\ &= \frac{1}{2} \left[e^{T} (I_{N} \otimes \bar{A}^{T} + \mathcal{L} \otimes K^{T} B^{T}) + \delta^{T} (I_{N} \otimes B^{T}) - \psi^{T} y^{T} (I_{N} \otimes B^{T}) \right] (\mathcal{L} \otimes P) e \\ &\quad + \frac{1}{2} e^{T} (\mathcal{L} \otimes P) \left[(I_{N} \otimes \bar{A} + \mathcal{L} \otimes BK) e + (I_{N} \otimes B) \delta + (I_{N} \otimes B) \psi y \right] \\ &= \frac{1}{2} e^{T} (I_{N} \otimes \bar{A}^{T}) (\mathcal{L} \otimes P) e + e^{T} (\mathcal{L} \otimes K^{T} B^{T}) (\mathcal{L} \otimes P) e \\ &\quad + \delta^{T} (I_{N} \otimes B^{T}) (\mathcal{L} \otimes P) e - \psi^{T} y^{T} (I_{N} \otimes B^{T}) (\mathcal{L} \otimes P) e \\ &\quad + e^{T} (\mathcal{L} \otimes P) (I_{N} \otimes \bar{A}) e + e^{T} (\mathcal{L} \otimes P) (\mathcal{L} \otimes BK) e \\ &\quad + e^{T} (\mathcal{L} \otimes P) (I_{N} \otimes B) \delta - e^{T} (\mathcal{L} \otimes P) (\mathcal{L} \otimes BK) e \\ &\quad + e^{T} (\mathcal{L} \otimes P) (I_{N} \otimes B) \delta - e^{T} (\mathcal{L} \otimes P) (I_{N} \otimes B) \psi y \\ &= \frac{1}{2} e^{T} \left[\mathcal{L} \otimes (\bar{A}^{T} P) + \mathcal{L}^{2} \otimes (K^{T} B^{T} P) + \mathcal{L} \otimes (P \bar{A}^{T}) + \mathcal{L}^{2} \otimes (P B K) \right] e \\ &\quad + \frac{1}{2} \delta^{T} \left[\mathcal{L} \otimes (B^{T} P) \right] e + \frac{1}{2} e^{T} \left[\mathcal{L} \otimes (P B) \right] \delta \\ &\quad - \psi^{T} y^{T} (\mathcal{L} \otimes B^{T} P) e - e^{T} (\mathcal{L} \otimes P B) \psi y \ . \end{split}$$

Since $K = -B^T P$, therefore $K^T = -PB$, and,

$$\dot{V} = \frac{1}{2}e^{T} \left[\mathcal{L} \otimes (\bar{A}^{T}P + P\bar{A}) - \mathcal{L}^{2} \otimes (PBB^{T}P + PBB^{T}P) \right] e \\ + \delta^{T} (\mathcal{L} \otimes B^{T}P)e - e^{T} (\mathcal{L} \otimes PB)\psi y .$$

From the Kalman-Yakubovich conditions (Kaufman et al., 1997) and the Riccati equation in Equation 5.5, $PB = \overline{C}^T$, where $y = \overline{C}e$, as defined before. Therefore,

$$B^T P = \overline{C} \longrightarrow y = B^T P e \longrightarrow y = e^T P B.$$

Continuing the derivation,

$$\dot{V} = \frac{1}{2}e^{T} \left[\mathcal{L} \otimes (\bar{A}^{T}P + P\bar{A}) - 2\mathcal{L}^{2} \otimes (PBB^{T}P) \right] e + \delta^{T} (\mathcal{L} \otimes y) - (\mathcal{L} \otimes y^{T}) \psi y$$
$$= \frac{1}{2}e^{T} \left[\mathcal{L} \otimes (\bar{A}^{T}P + P\bar{A}) - 2\mathcal{L}^{2} \otimes (PBB^{T}P) \right] e + (\mathcal{L} \otimes \delta^{T}y) - (\mathcal{L} \otimes y^{T}\psi y) .$$
(5.6)

Lemma 5.1. Suppose that the graph \mathcal{G} is both undirected and connected, and at least one agent has access to the leader, then \mathcal{L} is positive definite.

By Lemma 5.1, we know that \mathcal{L} is positive definite. Let U be an arbitrary matrix such that $U^T \mathcal{L} U = J$, and introduce a state transformation $\epsilon = (U^T \otimes I_N)e$. Equation 5.6 can then be written as,

$$\dot{V} = \frac{1}{2} \epsilon^{T} \left[J \otimes \left(\bar{A}^{T} P + P \bar{A} \right) - 2\alpha J^{2} \otimes \left(P B B^{T} P \right) \right] \epsilon + \left(\mathcal{L} \otimes \delta^{T} y \right) - \left(\mathcal{L} \otimes y^{T} \psi y \right)$$
$$= \frac{1}{2} \sum_{i=1}^{n} \lambda_{i} \epsilon_{i}^{T} \left(\bar{A}^{T} P + P \bar{A} - 2\alpha \lambda_{i} P B B^{T} P \right) \epsilon_{i} + \left(\mathcal{L} \otimes \delta^{T} y \right) - \left(\mathcal{L} \otimes y^{T} \psi y \right) . \quad (5.7)$$

If α is chosen sufficiently large, such that $2\alpha\lambda_i > 1$ (Z. Li et al., 2013), and it follows

that Equation 5.7 becomes,

$$\dot{V} \leq \frac{1}{2} \sum_{i=1}^{n} \lambda_{i} \epsilon_{i} \left(\bar{A}^{T} P + P \bar{A} - P B B^{T} P \right) \epsilon_{i} + \left(\mathcal{L} \otimes \delta^{T} y \right) - \mathcal{L} \otimes y^{T} \psi y .$$
 (5.8)

Using $\|\epsilon\|^2 = \|e\|^2$ and the following inequality (Balas & Frost, 2014; Khalil, 2002),

$$-\frac{1}{2}e^{T}Qe \le -\frac{1}{2}\lambda_{\min}(Q)\|e\|^{2},$$
(5.9)

Equation 5.8 becomes,

$$\begin{split} \dot{V} &\leq -\frac{1}{2} \min_{i=1,\dots,n}(\lambda_i) \underline{\sigma}(Q) \|e\|^2 + \left(\mathcal{L} \otimes \delta^T y\right) - \mathcal{L} \otimes y^T \psi y \\ &\leq -\frac{1}{2} \min_{i=1,\dots,n}(\lambda_i) \underline{\sigma}(Q) \|e\|^2 + \overline{\sigma}(\mathcal{L}) \overline{\sigma}(\delta^T y) - \overline{\sigma}(\mathcal{L}) \overline{\sigma}(y^T \psi y) \\ &\leq -\frac{1}{2} \min_{i=1,\dots,n}(\lambda_i) \underline{\sigma}(Q) \|e\|^2 + \overline{\sigma}(\mathcal{L}) \left[\delta^T y - y^T \psi y\right], \end{split}$$

where $\underline{\sigma}(Q) = \lambda_{\min}(Q)$ and $\overline{\sigma}(\mathcal{L}) = \lambda_{\max}(\mathcal{L})$.

The adaptive function $\psi(e, t)$ can be designed to be lower bounded as shown in Figure 5.1. The lower bound of the system adaptation gain yields,

$$\psi(e,t) = \eta f(\Delta u_x) \ge \eta \varepsilon = \overline{\gamma}. \tag{5.10}$$

Using the lower bound $\overline{\gamma}$ in Equation 5.8 gives,

$$\dot{V}(e,t) \le -\frac{1}{2} \min_{i=1,\dots,n} (\lambda_i) \underline{\sigma}(Q) \|e\|^2 + \overline{\sigma}(\mathcal{L}) \left[\delta(t)y - y^2 \overline{\gamma} \right].$$
(5.11)



Figure 5.1 System adaptive function (A. E. Rocha, 2016)

Completing the square using $\delta(t)y - y^2\overline{\gamma} = -\overline{\gamma}\left[y - \frac{\delta(t)}{2\overline{\gamma}}\right]^2 - \frac{\delta(t)^2}{4\overline{\gamma}}$ and simplifying, the inequality becomes,

$$\dot{V}(e,t) \leq -\frac{1}{2} \min_{i=1,\dots,n} (\lambda_i) \underline{\sigma}(Q) \|e\|^2 + \overline{\sigma}(\mathcal{L}) \left[-\frac{\delta(t)^2}{4\overline{\gamma}} \right]$$
$$\leq -\frac{1}{2} \min_{i=1,\dots,n} (\lambda_i) \underline{\sigma}(Q) \|e\|^2 - \frac{\overline{\sigma}(\mathcal{L}) \delta(t)^2}{4\overline{\gamma}} .$$
(5.12)

Remark. It is noted that since $\psi(e, t)$ is bounded, the error will be bounded in the absence of a disturbance δ . Setting $\delta(t) = 0$, Equation 5.12 becomes,

$$\dot{V}(e,t) \le -\frac{1}{2} \min_{i=1,\dots,n}(\lambda_i)\underline{\sigma}(Q) \|e\|^2 \le 0.$$
(5.13)

From the quadratic Lyapunov function, the following is true,

$$\frac{1}{2}\underline{\sigma}(P)\|e\|^2 \leq V(e) = \frac{1}{2}e^T(\mathcal{L}\otimes P)e \leq \frac{1}{2}\overline{\sigma}(P)\|e\|^2$$
(5.14)

therefore,

$$\|e\|^2 \le \frac{2V}{\overline{\sigma}(P)}.\tag{5.15}$$

Equation 5.12 can now be rewritten as,

$$\dot{V}(e,t) \leq -\min_{i=1,\dots,n}(\lambda_i)\frac{\underline{\sigma}(Q)}{\overline{\sigma}(P)}V + \frac{\overline{\sigma}(\mathcal{L})\delta(t)^2}{4\overline{\gamma}} \\
\leq -\tilde{K}V + \frac{\overline{\sigma}(\mathcal{L})\delta(t)^2}{4\overline{\gamma}},$$
(5.16)

where $\tilde{K} = \min_{i=1,\dots,n} (\lambda_i) \frac{\underline{\sigma}(Q)}{\overline{\sigma}(P)}$.

This differential equation is solved by multiplying both sides by $e^{\tilde{K}t}$, then,

$$e^{\tilde{K}t} \left[\dot{V} + \tilde{K}V \right] \le e^{\tilde{K}t} \frac{\overline{\sigma}(\mathcal{L})\delta(t)^2}{4\overline{\gamma}} .$$
(5.17)

Solving for V(t) yields,

$$V(t) \le V(0)e^{-\tilde{K}t} + e^{-\tilde{K}t} \int_0^\tau e^{\tilde{K}t} \frac{\overline{\sigma}(\mathcal{L})\delta(t)^2}{4\overline{\gamma}} d\tau.$$
(5.18)

The term $\delta^2(t)$ can be bounded by $\sup[\delta^2(t)] \ge \delta^2(t)$. This term is factorized from the

integral, and the inequality in Equation 5.14 can be reused,

$$\frac{1}{2}\underline{\sigma}(P)\|e\|^2 \le V(t) \le V(0)e^{-\tilde{K}t} + e^{-\tilde{K}t}\frac{\sup[\delta^2(t)]}{4\overline{\gamma}}\overline{\sigma}(\mathcal{L})\int_0^\tau e^{\tilde{K}t}d\tau.$$
(5.19)

Knowing $V(0) \leq \frac{1}{2}\overline{\sigma}(P) \|e(0)\|^2$ holds for the initial condition of the Lyapunov candidate, Equation 5.19 becomes,

$$\|e\|^{2} \leq \frac{2}{\underline{\sigma}(P)} V(0) e^{-\tilde{K}t} + \frac{\sup[\delta^{2}(t)]}{2\overline{\gamma}\underline{\sigma}(P)} \overline{\sigma}(\mathcal{L}) e^{-\tilde{K}t} \left[e^{\tilde{K}t} - 1 \right]$$
$$\|e\|^{2} \leq \frac{\overline{\sigma}(P)}{\underline{\sigma}(P)} \|e(0)\|^{2} e^{-\tilde{K}t} + \frac{\sup[\delta^{2}(t)]}{2\overline{\gamma}\underline{\sigma}(P)} \overline{\sigma}(\mathcal{L}) \left[1 - e^{-\tilde{K}t} \right]$$
$$\|e\| \leq \sqrt{\frac{\overline{\sigma}(P)}{\underline{\sigma}(P)}} \|e(0)\|^{2} e^{-\tilde{K}t} + \frac{\sup[\delta^{2}(t)]}{2\overline{\gamma}\underline{\sigma}(P)} \overline{\sigma}(\mathcal{L}) \left[1 - e^{-\tilde{K}t} \right]$$
(5.20)

Using $\sqrt{\|a\|^2 - \|b\|^2} \le \|a\| - \|b\|$, then,

$$\|e\| \le \sqrt{\frac{\overline{\sigma}(P)}{\underline{\sigma}(P)}} \|e(0)\| e^{-\tilde{K}t/2} + \sqrt{\frac{\sup[\delta^2(t)]\overline{\sigma}(\mathcal{L})}{2\overline{\gamma}\underline{\sigma}(P)}} \left[1 - e^{-\tilde{K}t/2}\right].$$
(5.21)

Evaluating the limit of $\sup(||e(t)||)$ and $\lim_{t\to\infty}$ on each side of Equation 5.21, the following region of convergence is obtained for the tracking error trajectories of the system with adaptive augmentation in the presence of bounded disturbances,

$$||e|| \leq \sqrt{\frac{\sup[\delta^2(t)]\overline{\sigma}(\mathcal{L})}{2\overline{\gamma}\underline{\sigma}(P)}} = \gamma_1 .$$
(5.22)

Summary: This chapter presented the stability proof for the proposed adaptive controller described in Chapter 4. A candidate Lyapunov function is proposed and derived. This derivative is then transformed through the use of Jordan decomposition and Kronecker product properties. In conclusion, the proposed controller is proven to be stable in the presence of external time-varying bounded disturbances for a MAS with an undirected connected communication graph. The results in Chapter 6 further show the controller's disturbance rejection capabilities for MASs under directed communication topology in specific cases.

6. DMRAC Numerical Simulations

The proposed distributed control architecture is tested under three different scenarios. Simulation results for the DMRAC are presented in this chapter. A first case shows how the DMRAC handles bounded disturbances. A second scenario refers to the performance of the DMRAC for compensating communication link cyber attacks. In a third case scenario, the effectiveness of the DMRAC against zero-dynamics attacks is studied.

Consider a network of four quadrotors described by Equation 3.47 with mass m = 1 kg, length l = 0.5 m and the acceleration of gravity g = 9.8 m/s². The directed communication topology is given in Figure 6.1. The initial states for each agent, such as the initial positions and attitude angles, are given in Table 6.1.



Figure 6.1 Information exchange topology

Results for consensus among agents are shown in Figure 6.2. The agents start at the positions listed in Table 6.1 and reach an agreement after approximately 20s. The gain values are listed in Table 6.2. The DMRAC ensures that all agents reach consensus, proving that the system is stable. This result serves as the benchmark for the results illustrated in the remainder of the chapter.

Table 6.1

Initial conditions

		States										
	x	y	z	ϕ	θ	ψ	<i>x</i>	\dot{y}	ż	$\dot{\phi}$	$\dot{ heta}$	$\dot{\psi}$
Agent 1	5	5	3	0	0	10	0	0	0	0	0	0
Agent 2	5	0	4	0	0	20	0	0	0	0	0	0
Agent 3	0	5	5	0	0	30	0	0	0	0	0	0
Agent 4	0	0	6	0	0	40	0	0	0	0	0	0

Table 6.2

Adaptive controller gain values

k_z	k_{ϕ}	$k_{ heta}$	k_ψ	$k_{\dot{z}}$	$k_{\dot{\phi}}$	$k_{\dot{\theta}}$	$k_{\dot{\psi}}$
30	30	1	30	10	30	1	10

6.1. Bounded Disturbances

To demonstrate the performance of the proposed adaptive controller used within the consensus algorithm for local disturbance rejection, a bounded disturbance chosen as $\delta = 10 \sin(\pi t)$ is injected in Agent 2 after 10s in the roll channel. These results are presented for the consensus control case (regulator) and for the formation control case (tracking). For these simulation test cases, the focus is on how the DMRAC responds to the onset and impact of specific disturbances; deterministic disturbance models are therefore preferred for the selected test cases.



Figure 6.2 Agents reaching an agreement on position

6.1.1. Consensus Control

The regulator case of the consensus control is presented in this section. The agents are to reach an agreement that depends only on the chosen initial conditions. Numerical simulations are run with the DMRAC off and on, as shown in Figure 6.4 and Figure 6.5, respectively. It can be observed from Figure 6.5 that all four agents reach consensus in spite of the bounded disturbance. The disturbance rejection in the roll in Agent 2 is successfully mitigated by the DMRAC.

Due to the chosen topology (Figure 6.1), the disturbances in an agent are also felt in the neighboring quadrotors. For example, Agent 4 receives information from Agent 3, and Agent 3 receives information from Agent 2. This explains why the roll disturbance is also felt by Agent 3 & 4 when the adaptation is off, then becomes undetectable once the adaptive controller is activated. Figure 6.10 shows how the adaptive inputs react to the disturbance injection at the 10s mark. In Agent 2, $u_{w_{AD}}$ is activated as soon as the disturbance is injected. This is expected since the disturbance affects the roll and $w = -g \tan \phi$.

It is also worth noting that the successful rejection of the disturbance is mainly due to the choice of reference model (Figure 6.3) within the adaptive controller. The consensus law is added to the linear feedback controller within the reference model to guarantee consensus of all the states. This strategy stabilizes the controller, thereby reducing the time spent on tuning the adaptive gains. Any further tuning goes to improve the effectiveness of the DMRAC.

6.1.2. Formation Control

The simulation results for the MAS under attack using a formation controller are presented. The case example is inspired from (Ren et al., 2007), where agents follow a reference trajectory all the while maintaining a spacing of 100m between the agents. The consensus protocol given in Equation 4.27 is then applied. A disturbance of $200 \sin(\pi t)$ is injected after 10 seconds in Agent 2; the outcome is shown in Figure 6.14.

A disturbance of $300 \sin(\pi t)$ is then injected after 10 seconds; the outcome is shown in Figure 6.15. The objective is to observe the impact of a larger disturbance, thus the same disturbance or a greater effect is injected after 10 seconds as shown in the x - y plot. The states are much more impacted and that in turn affects the neighbors to the point where the simulation eventually crashes.



Figure 6.3 Reference model: nominal conditions for DMRAC performance







Figure 6.5 Consensus with adaptation (baseline & adaptive controller)



















Figure 6.10 Evolution of the adaptive inputs for Agent 1 under bounded disturbance with and without AIS inspired adaptation law



Figure 6.11 Evolution of the adaptive inputs for Agent 2 under bounded disturbance with and without AIS inspired adaptation law



Figure 6.12 Evolution of the adaptive inputs for Agent 3 under bounded disturbance with and without AIS inspired adaptation law

Adaptive inputs for agent 3



Figure 6.13 Evolution of the adaptive inputs for Agent 4 under bounded disturbance with and without AIS inspired adaptation law

A similar observance is made in the consensus case, where the agents do not reach a consensus. In continuation, the simulation for formation consensus with and without the disturbance yields the same observation. In the case with the adaptation, the disturbance is completely compensated for and the trajectories remain unchanged.



Figure 6.14 Agent 2 is injected with a disturbance of $200 \sin(\pi t)$ after 10s of flight. Given the inherent robust nature of the distributed controller, the local controller (output in dashed lines) is sufficient to counter the attack. While it does manage to track the trajectory, the disturbance in Agent 2 impacts the desired distance and formation.

6.2. Communication Link Attack

Since the quadrotor MAS operates in a distributed consensus protocol, its communication network is vulnerable to cyber attacks such as communication link attacks. This type of attack, also known as false data-injection attack, tricks the agents into thinking that the new information they receive comes from the network.



Figure 6.15 Agent 2 is injected with a disturbance of $300 \sin(\pi t)$ after 10s of flight. Unlike in Figure 6.14, the local controller (output in dashed lines) is not sufficient to counter the attack. After the fault injection, the neighbors to Agent 2 are impacted until the entire network becomes unstable. The DMRAC (full lines) is effective.

The communication link attack can be implemented following the model reported in (Taheri et al., 2020). The development and outcome of this attack is shown in this section.

Assuming the attacker sends an attack signal $a_{\mathbf{x}}^{ji}(t) \in \mathbb{R}^n$ to agent *i*, the attack can be represented as,

$$a_{\mathbf{x}}^{j\imath}(t) = a_0 - \mathbf{x}_j(t),\tag{6.1}$$

where $a_0 \in \mathbb{R}^n$ is a constant vector and \mathbf{x}_j is the state vector of neighboring agent j. Agent i then receives the manipulated states $\mathbf{x}_a^{ji}(t) = \mathbf{x}_j(t) + a_{\mathbf{x}}^{ji}(t)$ from agent j and is now under cyber-attack.

6.2.1. Consensus Control

The closed-loop dynamics for the MAS under communication-link attack is represented as,

$$\dot{\mathbf{x}}_i = A\mathbf{x}_i + BK_1\mathbf{x}_i - BK_2\sum_{j\in\mathcal{N}_i}(\mathbf{x}_i - \mathbf{x}_j) - BK_{AD}(\mathbf{x}_i - \mathbf{x}_{m_i}) - Bq_if_i , \qquad (6.2)$$

where,

$$q_i = \begin{cases} 1, & \text{if link } (i, j) \text{ is under attack;} \\ \\ 0, & \text{otherwise,} \end{cases}$$
(6.3)

and,

$$f_i = \sum_{j \in \mathcal{N}_i} a_{\mathbf{x}}^{ji} , \qquad (6.4)$$

is the attack on all incoming communication links of agent i.

When a link is under attack, Equation 6.2 becomes,

$$\dot{\mathbf{x}}_i = A\mathbf{x}_i + BK_1\mathbf{x}_i - BK_2\sum_{j\in\mathcal{N}_i}(\mathbf{x}_i - a_0) - BK_{AD}(\mathbf{x}_i - \mathbf{x}_{m_i}) .$$
(6.5)

The information from the neighbors is blocked and supplanted with an attack signal a_0 . The choice of a_0 determines the strength of the attack. For example, if $a_0 = [0 \ 0 \ 0]^T$, the attack is a DoS attack and the closed-loop dynamics for the agent under attack becomes,

$$\dot{\mathbf{x}}_i = A\mathbf{x}_i + BK_1\mathbf{x}_i - BK_2\sum_{j\in\mathcal{N}_i}\mathbf{x}_i - BK_{AD}(\mathbf{x}_i - \mathbf{x}_{m_i}) .$$
(6.6)
In order to study the effect of the communication link attack on a MAS, different tests are performed to observe the extent of the attack's aftermath. The attacks are injected into one agent from the MAS under the directed topology given by Figure 6.1, with varying strengths in the attack.



Figure 6.16 3D position of agents' trajectories after communication link attack. After initially reaching an agreement on the position, the agents are met with a communication link attack. The new information is treated as a new agreement value, and the team agrees on a new consensus point, (200, 2, 4)m in this case.

The communication link attack is injected in Agent 1 after 20s. A first test is performed for 100s with $a_0 = [50 \ 50 \ 50]$. The time of the attack is represented by a black mark on the plots in Figure 6.16. Before the attack injection, the agents do reach an agreement. This can be seen in Figure 6.18, where the agents reach an agreement for the x-direction at 3m. After the attack, the agents need to agree on a new agreement value, which now becomes 200m. The adaptive controller ensures that the agent states reach an agreement even though it does not compensate for this type of attack.

6.2.2. Formation Control

To study the effect of the communication link attack on a MAS during formation, different tests are performed. The attacks are injected in one agent, for different communication topologies (undirected and directed), with varying strengths in the attack. The study is limited to the agents following a specific trajectory while maintaining a linear formation.

The same test setup as for the consensus control is applied for the attack on the formation. The communication link attack is injected in Agent 1 after 20s. The time of the attack is represented by a black mark on the plots in Figure 6.16. The dashed lines show the agent's trajectory with the baseline controller only, the solid lines are the agent trajectories with the DMRAC in action.

From the plots, it is noticeable that the DMRAC keeps the agents stable even when the communication link attack is strong as in Figure 6.19. When the attack is not as disruptive, the baseline controller and the adaptive controller have a comparable performance as shown in Figure 6.19. Although the attacked agent is initially disoriented, it eventually gets back on track and attempts to follow the formation from its reference model.

In the plots shown in Figure 6.20, a communication link attack is injected in each agent in the network (one agent under attack per test) to better visualize the effects of the attack on the rest of the agents. The adaptive controller results (bold







Figure 6.18 Agents states trajectories after communication link attack (zoom). In continuation with Figure 6.16, the states trajectories for the agents are amplified. The zoomed-in section confirms how the agents had reached an initial agreement prior to the attack. The adaptive controller and the local controller are able to keep the system stable by ensuring that consensus is met, despite the disruption to the original agreement.

lines) are compared with the reference trajectories (dotted lines) for each agent.

Irrespective of which communication link is under attack, the outcome of this cyber attack on the MAS is the same in all test cases since the agents are negatively impacted by the attack. While this result is expected, this confirmation could lead to developing controllers capable of mitigating this type of attack.

6.3. Zero-Dynamics Attack

The zero-dynamics attack is a stealthy and aggressive attack. This type of attack is unbounded (Figure 6.21), and very challenging to detect and compensate for. In studying the ZDA in this research, the goal is specifically to evaluate whether the proposed control architecture can slow down the effect of the ZDA on the agent



Figure 6.19 Effect of communication link attack on formation keeping with undirected topology when Agent 1 states are under attack. Agent 1 receives a communication link attack while the team must keep a formation. The fault is injected after 20s (or when the agents reach the \times). By itself, the local controller is no longer able to keep the formation and goes unstable after some time. Though the adaptive controller does not perfectly keep the formation, it does compensate for the attack and guides the agent in the general direction of the formation.



Figure 6.19 (cont.) Effect of communication link attack on formation keeping with undirected topology when Agent 1 states are under attack. Unlike in the previous test cases, both the local and the adaptive controller now give similar outputs for the formation control of the MAS under attack. This is because the values taken on by a_0 are very similar to the states communicated along these channels.



Figure 6.20 Effect of communication link attack on formation with directed topology when an agent is under attack.



Figure 6.20 (cont.) Effect of communication link attack on formation with directed topology when an agent is under attack.

dynamics so that the ZDA can be detected by the global self health monitoring system presented in Chapter 7.



Figure 6.21 Example of an unbounded disturbance generated for a zero-dynamics attack. In this case, the unbounded disturbance is undetectable for many seconds, but due to the exponential that governs it, it rapidly becomes unstable.

Consider a linear system defined by a strictly proper scalar transfer function that does not have any common zeros and poles,

$$g(s) = \frac{p(s)}{d(s)} = k \frac{s^m + p_1 s^{m-1} + \dots + p_{m-1} s + p_m}{s^n + d_1 s^{n-1} + \dots + d_{n-1} s + d_n} , \qquad (6.7)$$

where k is the input gain. The roots of the polynomial p(s) are called *transmission* zeros of the system. The system in Equation 6.7 can be written in state-space representation (controller canonical form) as,

$$\dot{x} = Ax + Bu \tag{6.8}$$
$$y = Cx$$

where,

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -d_n & -d_{n-1} & -d_{n-2} & \dots & -d_1 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ \vdots \\ k \end{bmatrix}, \qquad C = [p_m \dots p_1 \ 1 \ 0 \dots \ 0].$$

The zeros of the transfer function correspond to the poles of the zero dynamics (or internal dynamics) when the output is identical to zero for a finite interval. When the zero dynamics of a linear system are stable (zeros in the left-hand plane), the system is *minimum phase*. If a plant has unstable internal dynamics that are not visible at the system's output, it will generate destabilizing effects in the system. The zero-dynamics of a nonlinear system,

$$\dot{x} = f(x) + g(x)u$$

$$(6.9)$$

$$y = h(x) ,$$

are the reduced-order dynamics resulting from forcing the output to be zero for all times.

To inject the dynamic attack, a transfer function of the system is chosen after selecting the input which is to be affected by the ZDA. In this case, an attack is performed on the longitudinal dynamics,

$$\frac{u'_3}{\theta} = \frac{9.8}{s^4 + 10s^3 + 35s^2 + 50s} \ . \tag{6.10}$$

This transfer function is then discretized and its smallest zero is selected as the attack signal to be injected back into the system. The discretized transfer function is given by,

$$\frac{u'_3}{\theta} = \frac{1.593 * 10^{-6} z^2 + 6.215 * 10^{-6} z + 1.515 * 10^{-6}}{z^3 - 2.901 z^2 + 2.806 z - 0.9048} .$$
(6.11)

The zeros of the discretized system are $z_1 = -3.64$ and $z_2 = -0.2613$. The attack transfer function is chosen as,

$$G_{ZDA} = \frac{1}{z + 3.64} \ . \tag{6.12}$$

A representation of the simulation environment in which the ZDA creation and injection takes place is shown in Figure 6.22.



Figure 6.22 ZDA injection in simulation environment

6.3.1. Consensus Control

The zero-dynamic attack from both the continuous and discrete point of view is presented in Figure 6.23. In this case, both plots are shown together and it can be observed that in the continuous time, in real time, the system goes unstable in a fraction of a second. The quadrotor is already unstable, but this is not yet detected from the computer's perspective in the discrete time. It takes five sample times before the onboard computer would be able to detect that there has been an attack, by which time the system has already failed. This demonstrates the stealthiness of the attack.



Figure 6.23 Zero dynamic attack. The continuous and discrete outputs are plotted together to illustrate the stealthiness of the ZDA. While the onboard computer (discrete-time output) has barely detected the attack, the system (continuous-time output) has already gone unstable.

Summary: This chapter presented simulation results in three parts: bounded sinusoidal disturbances, communication link attacks and zero-dynamics attacks. It demonstrated the output for both the consensus and formation controller in each of the three case scenarios. The presented results show that the proposed DMRAC is successfully able to keep the system stable when facing different types of bounded disturbances.

7. Global Self Numerical Simulations

In expanding MAS network capabilities, this chapter introduces the development of the immunity-based methodology for fault detection within a network of agents. It is discussed at its initial stage and is herein referred to as Global Health Monitoring (GHM). This work expands the application of the AIS paradigm from a single agent, the local level, to an entire network of agents, the global level. Preliminary numerical simulations are presented with an analysis of the health management capacity to detect failures at the global level.

Following on Chapter 4, the AIS paradigm has been used as an intelligence based strategy in areas such as anomaly detection, data mining, computer security, adaptive control, and pattern recognition. It has diagnosis and prognosis capabilities that operate in a like manner as does the biological immune system, sourced from the principle of self/nonself discrimination, when it detects exogenous antigens while not reacting to the self-cells. Recent tests conducted by the research team at the Advanced Dynamics and Control Laboratory (ADCL) have demonstrated the AIS capabilities to timely detect, identify and compensate for abnormal conditions in aerospace vehicles that might drive a system outside of its bounds of nominal design (Garcia et al., 2016). This is the context in which the work presented in this chapter is inscribed.

7.1. Generation of Self and Non-Self Hyperspace

The steps in the development of an AIS are illustrated in this section. The starting point is the selection and definition of desired *features*. Features are identified as the variables (in general, functions of time t) that fully define the chosen

system and are most likely to distinguish the traits of the selected abnormal conditions. These traits can be defined in terms of occurrence, presence, type, severity, and consequences.

When the artificial immune system is applied to system fault detection, it is able to detect Abnormal Conditions (ACs) whenever the system detects a change in the standard configuration of features, in other words, when a configuration is different from a predetermined set. The set is trusted as nominal and functions in normal situations. Hence, a failure in one agent (or subsystem) is perceived as an intrusion by antigens (Moncayo et al., 2011). In this setup, the feature values have a similar role as the biological chemical markers as located on antibody paratopes and have the function of encoding the self/non-self (Perhinschi & Moncayo, 2018). Thus, at any one time, a selected set of features can include information that is pertinent to the system's behavior and can also capture new or familiar abnormal situations. Examples of features include measured and estimated states, sensor outputs and statistical parameters.

As a data-driven methodology, the AIS paradigm requires a considerable set of test data to determine the self or the features in normal conditions. The paradigm uses large numerical representations of the self/non-self and this data is then processed and stored within the hardware. If the representation of the self is built correctly, then the self can successfully detect the fault.

7.1.1. Self/Non-Self Representation

A set \mathcal{F} of all features φ_i within a N-dimensional real hyperspace $\mathcal{U} \subset \mathbb{R}^N$ (referred to as *Universe*), contains features selected across the entire MAS,

$$\mathcal{F} = \{\varphi_i \mid i = 1, 2, \dots, N\},\tag{7.1}$$

where N is the total number of features. A *feature point* P is the N-dimensional vector of all simultaneous values at any given instant \bar{t} of features φ_i ,

$$P = [\varphi_{1P} = \varphi_1(t = \bar{t}) \quad \varphi_{2P} = \varphi_2(t = \bar{t}) \quad \dots \quad \varphi_{NP} = \varphi_N(t = \bar{t})] \quad , \quad P \in \mathcal{U}.$$
(7.2)

The features are typically normalized based on known or estimated reference values under abnormal conditions to take values between 0 and 1,

$$\varphi_i \in \{0, \dots, 1\} . \tag{7.3}$$

The point \mathcal{O} with coordinates $[\varphi_1 = 0, \varphi_2 = 0, \dots, \varphi_N = 0]$ is considered the origin of an orthogonal coordinate system $\overline{\mathcal{U}}$ associated to the hyperspace \mathcal{U} . Therefore, the Universe of interest becomes a hyper-cube of unit side and the feature point P can be represented by its position vector with respect to O, \vec{r}^{OP} , whose coordinates with respect to $\overline{\mathcal{U}}$ are denoted as,

$$\left[\vec{r}^{OP}\right]_{\overline{\mathcal{U}}} = \begin{bmatrix} \varphi_{1P} & \varphi_{2P} & \dots & \varphi_{NP} \end{bmatrix} .$$
(7.4)

The self S is defined as the hyper-subspace of all possible feature points at normal conditions and all other points in \mathcal{U} form the non-self \overline{S} . Therefore,

$$\mathcal{S} \cup \bar{\mathcal{S}} = \mathcal{U} \quad \text{and} \quad \mathcal{S} \cap \bar{\mathcal{S}} = \emptyset$$
. (7.5)

For practical reasons, the self points are clustered and the self S is represented as a set S of hyper-bodies including these clusters. The non-self \overline{S} is covered with similar hyper-bodies to produce a set of non-self clusters \overline{S} . Different shapes can be considered for the self/non-self representation, such as hyper-cubes, hyper-spheres and hyper-ellipsoids (Perhinschi & Moncayo, 2018). The hyper-spherical representation is chosen in this study for the GHM development. For the case with N_c clusters c_i , the self can be expressed as,

$$\mathcal{S} = \{ c_1 \quad c_2 \quad \dots \quad c_{N_c} \}, \tag{7.6}$$

and the self clusters as,

$$c_{i} = \begin{bmatrix} C_{i} & R_{c_{i}} \end{bmatrix}$$
$$= \begin{bmatrix} \varphi_{1i} & \varphi_{2i} & \dots & \varphi_{Ni} & R_{c_{i}} \end{bmatrix},$$
(7.7)

where C_i is the center and R_{c_i} is the radius of the self cluster *i*. In this hyper-spherical representation with $N_{\bar{c}}$, the non-self clusters \bar{c} are generated in a similar manner. The non-self clusters will be called the *detectors*. When the detectors carry information, they then become identifiers usable for AC identification purposes. The following set of projections is a representation of the self/non-self given by the sub-selves π_i ,

$$\mathcal{S} = \{ \pi_1 \quad \pi_2 \quad \dots \quad \pi_N \} \,, \tag{7.8}$$

where the representation of each sub-self may consist of a set of clusters. For example,

$$\pi_i = \{ c_{i1} \quad c_{i2} \quad \dots \quad c_{iN_c} \} \,, \tag{7.9}$$

$$c_{ij} = \begin{bmatrix} C_{ij} & R_{c_{ij}} \end{bmatrix} = \begin{bmatrix} \varphi_{1ij} & \varphi_{2ij} & \dots & \varphi_{Nij} & R_{c_{ij}} \end{bmatrix} .$$
(7.10)

Table 7.1 illustrates the key definitions for the AIS paradigm. Looking at the characteristics of hyperspaces relative to distances and thresholds, it is noteworthy to assume that when the dimensionality of the hyperspace goes to infinity, the volume of the unit hyper-cube (the Universe) remains equal to one, while the volume of the inscribed hyper-sphere goes to zero. As a result, distances to border hyper-planes of the hyper-cube may be infinitesimal, whereas distances to hyper-cube corners may be extremely large. This means that the intuition in establishing thresholds and assessing distances, which is built in the 3D physical space, becomes inoperational. The impact of this counter-intuitive effect is felt when the number of features is higher than 10.

7.1.2. Hierachical Multiself Strategy

The Hierachical Multiself Strategy (HMS) has been previously proposed to mitigate these effects (Moncayo et al., 2011). Whereas all features are used to capture various types of ACs, the observation is that only a certain number of subsets may be needed to capture the various forms of any individual AC.

In the process, the sets of features are placed as subsets that define the sub-selves. The latter is then evaluated for its capacity to capture selected ACs. This information is ranked in a decision logic process. The outcome is based on each sub-self ranking in capturing the ACs, and a composition logic, which can then be formulated for detection, identification, and evaluation. If the dimension of the projections is larger or equal to the number of the ACs, then the performance penalty when using lower dimensional projections may be reduced to zero. In practice, even if hidden regions of the non-self exist and are reached with low probability, the overall discrimination performance may still be good.

Table 7.1

Main biological terms $\!\!\!\!^*$

Biological Term	AIS Paradigm Counterpart
Self or host organism	Agents under normal conditions;
	Sets of feature clusters and/or their projec-
	tions under normal operating conditions
Non-self or alien entities	Regions of the feature hyperspace that are
	unreachable or reachable under abnormal
	conditions.
	Sets of complementary feature clusters
	and/or their projections outside of normal
	operating conditions
Antigen	Set of current feature values (feature points)
	at abnormal conditions
Antibody	Data cluster in the non-self feature hyper-
	space (detector)
Organic markers (proteins and	System features or characteristic variables
other compounds)	and their values
Innate response	Baseline (local) controller
Adaptive response	Adaptive controller

*Certain terms adapted from Perhinschi & Moncayo (2018)

7.1.3. Antibody Generation

In antibody generation, the approach used in this work is based on a process referred to as a raw data set union (RDSU) method that processes experimental data at normal conditions. First, it combines different flight test samples in one single data file. Nominal flight data collection must take place in supervised and controlled conditions that most accurately represent ideal flying conditions. Failure data collection requires multiple flight tests where different failures or disturbance scenarios are played out.

In the work presented in this chapter, the data is gathered from numerical simulations. The DMRAC code for formation tracking is used for the nominal flight data collection. The failure data is obtained from injecting an agent in the MAS with a sinusoidal bounded disturbance and turning off the adaptation law.

Based on the span of the flight data and a percentage margin, this data file is then normalized. This is followed by the elimination of any duplicated points within the generated hyperspace to decrease the amount of storage. An optimized algorithm is used to cluster the reduced data. Data for each sub-self are processed separately to produce a set of antibodies by covering the lower dimensional non-self. Lastly, the generated self-clusters are used to generate antibodies through a negative selection-type process by covering the non-self hyperspace. The antibodies generation process can be stopped after the chosen number of iterations, either when the maximum number of acceptable detectors is reached, or when the desired coverage of the non-self is obtained.

7.2. Fault Detection in Quadrotor MAS

The AIS paradigm presented in Section 7.1 is used as a tool to detect faults across agents within a distributed network. The idea is to provide the network with a sense of global immunity, where an agent or a group of agents can sense anomalous behavior patterns in their neighbors. Not only would this act as a backup fault detection tool for the faulty agent, but this system would also ensure that the agents in the group are aware of the current health condition of their neighbors. With this information at hand, the agents would then be capable of making new global decisions that would benefit the mission collectively, such as trajectory re-planning or even formation adjustment. When combined with each individual agent's own HMS, this would help provide a more accurate diagnosis.

The study progresses on to the generation of detectors in a two-dimensional space for the global immune system. The same MAS topology used in Chapter 6 is used throughout this section. Four agents are in a directed topology which is described by the following Laplacian matrix,

$$\mathcal{L} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & 0 & -1 \\ -1 & -1 & 2 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}.$$
 (7.11)

7.2.1. 2D Detectors

In this case study, 2D detectors are first generated using the steps described in Section 7.1. The chosen features are,

$$\{x_1, y_1, \phi_1, \theta_1, \dots, x_n, y_n, \phi_n, \theta_n\}.$$
 (7.12)

The features $\{x, y, \phi, \theta\}$ are selected for all n = 4 agents, and the total number of features is 4n = 16. The maximum number of feature (f) combinations to generate

two-dimensional projections (p) is given by the binomial coefficient,

$$N_{\text{self}} = C_f^p = \binom{f}{p} = \frac{f!}{p!(f-p)!}$$

which in this case yields $C_{16}^2 = \frac{16!}{2!(14)!} = 120$ possible projections.

Remark: For MASs, it is not necessary to select the same number of features for all agents. Features can also be selected only from certain agents in the network. The goal is to have features that will create a unique representation of the MAS in order to detect specific ACs.

Of the 120 possible projections (Table 7.2), 20 were analyzed. In Figure 7.1, a 2D projection example of the hyperspace is shown along the x position of Agent 2, x_2 , and the roll angle of Agent 3, ϕ_3 . The red detectors are generated all around the self data points. To evaluate the effectiveness of this projection, data corresponding to a flight with an injected disturbance are projected along side the nominal data (the failure data is shown in black). In this particular case, the failure data is distinct from the self data. Furthermore, its trajectory goes through multiple detectors, indicating a high chance of fault detection when using these detectors during a flight.

In some test cases, the projections are unusable, as in the case with the detectors in Figure 7.2 where the self and the nonself data are indistinguishable, and the nonself data does not activate any detectors. These detectors are therefore not considered for fault detection.

In this scenario, the 2D view limits the analysis of the data since it can only give the projection between the states of two agents. To maximize the information

Table 7.2

Self	Fea	tures	Self	Fea	tures	Self	Fea	tures	Self	Fea	tures
1.	x_1	y_1	31.	ϕ_1	x_2	61.	x_2	$ heta_3$	91.	θ_2	ϕ_4
2.	x_1	ϕ_1	32.	ϕ_1	y_2	62.	x_2	x_4	92.	θ_2	$ heta_4$
3.	x_1	$ heta_1$	33.	ϕ_1	ϕ_2	63.	x_2	y_4	93.	x_3	y_3
4.	x_1	x_2	34.	ϕ_1	θ_2	64.	x_2	ϕ_4	94.	x_3	ϕ_3
5.	x_1	y_2	35.	ϕ_1	x_3	65.	x_2	$ heta_4$	95.	x_3	$ heta_3$
6.	x_1	ϕ_2	36.	ϕ_1	y_3	66.	y_2	ϕ_2	96.	x_3	x_4
7.	x_1	θ_2	37.	ϕ_1	ϕ_3	67.	y_2	θ_2	97.	x_3	y_4
8.	x_1	x_3	38.	ϕ_1	$ heta_3$	68.	y_2	x_3	98.	x_3	ϕ_4
9.	x_1	y_3	39.	ϕ_1	x_4	69.	y_2	y_3	99.	x_3	$ heta_4$
10.	x_1	ϕ_3	40.	ϕ_1	y_4	70.	y_2	ϕ_3	100.	y_3	ϕ_3
11.	x_1	$ heta_3$	41.	ϕ_1	ϕ_4	71.	y_2	$ heta_3$	101.	y_3	$ heta_3$
12.	x_1	x_4	42.	ϕ_1	$ heta_4$	72.	y_2	x_4	102.	y_3	x_4
13.	x_1	y_4	43.	$ heta_1$	x_2	73.	y_2	y_4	103.	y_3	y_4
14.	x_1	ϕ_4	44.	$ heta_1$	y_2	74.	y_2	ϕ_4	104.	y_3	ϕ_4
15.	x_1	$ heta_4$	45.	$ heta_1$	ϕ_2	75.	y_2	$ heta_4$	105.	y_3	$ heta_4$
16.	y_1	ϕ_1	46.	$ heta_1$	θ_2	76.	ϕ_2	θ_2	106.	ϕ_3	$ heta_3$
17.	y_1	$ heta_1$	47.	$ heta_1$	x_3	77.	ϕ_2	x_3	107.	ϕ_3	x_4
18.	y_1	x_2	48.	$ heta_1$	y_3	78.	ϕ_2	y_3	108.	ϕ_3	y_4
19.	y_1	y_2	49.	$ heta_1$	ϕ_3	79.	ϕ_2	ϕ_3	109.	ϕ_3	ϕ_4
20.	y_1	ϕ_2	50.	$ heta_1$	$ heta_3$	80.	ϕ_2	$ heta_3$	110.	ϕ_3	$ heta_4$
21.	y_1	θ_2	51.	$ heta_1$	x_4	81.	ϕ_2	x_4	111.	θ_3	x_4
22.	y_1	x_3	52.	$ heta_1$	y_4	82.	ϕ_2	y_4	112.	$ heta_3$	y_4
23.	y_1	y_3	53.	$ heta_1$	ϕ_4	83.	ϕ_2	ϕ_4	113.	$ heta_3$	ϕ_4
24.	y_1	ϕ_3	54.	$ heta_1$	$ heta_4$	84.	ϕ_2	$ heta_4$	114.	θ_3	$ heta_4$
25.	y_1	$ heta_3$	55.	x_2	y_2	85.	θ_2	x_3	115.	x_4	y_4
26.	y_1	x_4	56.	x_2	ϕ_2	86.	θ_2	y_3	116.	x_4	ϕ_4
27.	y_1	y_4	57.	x_2	$ heta_2$	87.	θ_2	ϕ_3	117.	x_4	$ heta_4$
28.	y_1	ϕ_4	58.	x_2	x_3	88.	θ_2	$ heta_3$	118.	y_4	ϕ_4
29.	y_1	$ heta_4$	59.	x_2	y_3	89.	θ_2	x_4	119.	y_4	$ heta_4$
30.	ϕ_1	$ heta_1$	60.	x_2	ϕ_3	90.	θ_2	y_4	120.	ϕ_4	$ heta_4$

List of all considered 2D self projections

available from all the neighbors of the faulty agent, detectors of a higher dimension are also generated as illustrated in the next section.



Figure 7.1 A 2D projection example of the hyperspace shown along the x position of Agent 2, x_2 , and the roll angle of Agent 3, ϕ_3 . The failure data (black) are distinct from the nominal data (blue) and falls within the detectors (red). This is a desired projection of the features since the detectors will be activated during abnormal conditions and minimize false alarms.

7.2.2. High-Order Detectors

One area of attention when generating the 2D detectors to determine faults at the MAS level is the high number of selves that would be generated. This aspect can make the 2D self representation impractical for a high number of agents. The problem arises at the feature definition step where the more a network has agents, the



Figure 7.2 2D projection using x_2 vs x_3 . Unlike in Figure 7.1, the nonself falls exactly within the space of the self data. The detectors are not activated even though the MAS is experiencing abnormal conditions. This is an example of a *false negative* detection where failure conditions can fallaciously be declared as nominal. The projection is not desired and will not be used as part of the GHM scheme.

more 2D projections will need to be analyzed and processed. In an effort to analyze varied data sets across multiple agents at once, the proposed solution is to increase the dimension of the features. Higher-order detectors (dim \geq 3) are therefore proposed for this case. The goal is to create high-order selves to better capture subtle changes in the agents during flight. The selected 4D selves are listed in Table 7.3,

where Selves 1 - 4 are composed of features from all agents and Selves 5 - 8 correspond the respective immune system of each agent. A new feature, δ_i , is the average distance between agent *i* and its neighbors and is given by,

$$\delta_i = \sum_{j=1}^n a_{ij} \sqrt{\|x_i - x_j\|^2 + \|y_i - y_j\|^2}, \quad j \in \mathcal{N}_i.$$
(7.13)

This new feature replaces the position features x_i and y_i previously used in the 2D selves, making the selves dependent on the relative positions between the agents and their neighbors given by the topology. In this configuration, the generated high-order selves will detect ACs at the global level so long as the topology is invariant.

Table 7.3

Selves		Features					
Self 1	δ_1	ϕ_2	$\dot{\phi}_3$	u_{w_4}			
Self 2	ϕ_1	$\dot{\phi}_2$	u_{w_3}	δ_4			
Self 3	$\dot{\phi}_1$	u_{w_2}	δ_3	ϕ_4			
Self 4	u_{w_1}	δ_2	ϕ_3	$\dot{\phi}_4$			
Self 5	δ_1	ϕ_1	$\dot{\phi}_1$	u_{w_1}			
Self 6	δ_2	ϕ_2	$\dot{\phi}_2$	u_{w_2}			
Self 7	δ_3	ϕ_3	$\dot{\phi}_3$	u_{w_3}			
Self 8	δ_4	ϕ_4	$\dot{\phi}_4$	u_{w_4}			

Generated high-order selves (part a)

Various initial tests are conducted with a view to determine the efficacy of the GHM setup for the MAS. The scenario uses a sinusoidal disturbance $(250 \sin \pi t)$ injected in Agent 1 from 30s to 50s, as illustrated in Figure 7.3. The activated antibodies (y-axis) are shown in time (x-axis). The values on the y-axis corresponds

to the specific detectors that are activated, and not the total number of activated AB.

In this case, the fault is detected shortly after it occurs by the detectors in Self 1 (Figure 7.3a) and Self 2 (Figure 7.3b). This is evidenced by the activation of some detectors a few seconds later for each self representation, where each circle represents an instance when the AB is activated. While Self 1 detectors are activated shortly after the fault injection, they do not continue to signal the fault after its initial detection. However, Detectors 90, 140 and 208 - 210, among others in Self 2, are capable of detecting the same injected fault for the duration of the upset conditions. For this specific fault, Self 2 detectors would be favored for AC detection.

7.2.3. Detection Rates and False Alarms

When using the antibodies as detectors, careful consideration should be given to detectors that generate false alarms (FAs). These detectors will be activated even though no AC has occurred. The detection rate (DR) and false alarm can be computed as,

$$DR = \frac{TP}{TP + FN} \times 100, \qquad FA = \frac{FP}{FP + TN} \times 100, \tag{7.14}$$

where, TP, TN, FP and FN represent different conditions of the detection logic:

- True Positive (TP): A failure is detected and declared as failure,
- True Negative (TN): Nominal conditions are declared as nominal,
- False Positive (FP): Nominal conditions are declared as failures,
- False Negative (FN): Failure condition is not detected.



(a) Antibody activation in Self 1 with fault in Agent 1



(b) Antibody activation in Self 2 with fault in Agent 1

Figure 7.3 AB activation with a fault in Agent 1. Figure 7.3a shows the Self 1 detectors that are activated during the simulation. The fault occurs at 30s, some detectors are activated a few seconds after. The value on the y-axis corresponds to the specific detectors that are activated, and not the total number of activated AB.

In an effort to reduce the number of false alarms, detectors that generate FAs must be discarded and not used within the detection process. In Figure 7.3, the antibodies generating false alarms (continuous bold lines) are activated during the entirety of the flight.

Similar tests are run for the other agents in the network using the selves given in Table 7.3 and the results are shown in Figure 8.1. Even though some FA rates appear to be high, this can be avoided by excluding the ABs responsible for the FAs, as previously explained. Bearing this in mind, the detection results for Agents 1 & 4 are promising.

7.2.4. Global Detection of Faulty Agent

In the following detection example, twelve selves are generated from features taken from all agents in the network. A communication topology is used to match one that would work for the agents in the formation depicted in Figure 7.4.



Figure 7.4 New communication topology for a formation keeping case

The examples shown in Figure 7.5 to Figure 7.7 are the fault detection results for a fault in Agent 1. A sinusoidal disturbance $(250 \sin \pi t)$ is injected in Agent 1 from 10s to 30s. The selves used are detailed in Table 7.4. On the one hand, Self 10 stands out as an example of a desirable DR, with low FA. On the other hand, Selves 5 & 9 stand out due to their high DR with high FA. The ABs that initially caused high FA rates are identified and manually removed from the selves. Following this procedure, it is noticed that Self 5 still has a relatively high DR, while the DR for Self 9 significantly drops. The detection rates and false alarms for each self are given in Figure 7.8.

Table 7.4

Selves	Features					
Selves $1 - 8$	see Table	7.3				
Self 9	δ_1	δ_2	δ_3	δ_4		
Self 10	ϕ_1	ϕ_2	ϕ_3	ϕ_4		
Self 11	$\dot{\phi}_1$	$\dot{\phi}_2$	$\dot{\phi}_3$	$\dot{\phi}_4$		
Self 12	$ $ u_{w_1}	u_{w_2}	u_{w_3}	u_{w_4}		

Generated high-order selves (part b)



Figure 7.5 Fault detection in Agent 1 by the ABs in Self 10. Self 10 ABs have an excellent DR and a low percentage of false alarms (40s - 50s). The result is further improved once the ABs that cause FAs are discarded (Figure 7.8).



(b) Antibody activation (without FA) in Self 5

Figure 7.6 This self corresponds to Agent 1's own immune system. Various ABs are activated during the fault injection from 10s until 30s.



(b) Antibody activation (without FA) in Self 9

Figure 7.7 Fault detection in Agent 1 by the AB in Self 9. The ABs that initially caused high FA rates are identified and manually removed from the self. DR and FA changes are mapped in Figure 7.8.



(b) DR & FA after removing the FA-causing ABs

Figure 7.8 DR & FA in Selves 1 - 12. The detection rate percentage is shown along with the corresponding percentage of FA for each self. The ABs that initially caused high FA rates are identified and manually removed from the selves. The procedure drastically improves the detection rates while lowering the false alarms.

In the test previously shown, features from the faulty agent were included as part of the global features. In this next configuration, only features from the other agents in the network are used for the purpose of testing global detection capabilities. To illustrate this, the following example is designed to detect a fault in Agent 1. For this reason, the selves given in Table 7.5 only include features from Agents 2 - 4. Using the same topology and disturbance model as in subsection 7.2.4., the formation consensus case is run and the detection results are analyzed hereafter.

Table 7.5

Selves for global fault detection in Agent 1: Selves $\{1 - 4; 8 - 11\}$ are 3D, Selves $\{5 - 7\}$ are 4D.

Selves	Features					
Self 1		ϕ_2	$\dot{\phi}_3$	u_{w_4}		
Self 2		$\dot{\phi}_2$	u_{w_3}	δ_4		
Self 3		u_{w_2}	δ_3	ϕ_4		
Self 4		δ_2	ϕ_3	$\dot{\phi}_4$		
Self 5	δ_2	ϕ_2	$\dot{\phi}_2$	u_{w_2}		
Self 6	δ_3	ϕ_3	$\dot{\phi}_3$	u_{w_3}		
Self 7	δ_4	ϕ_4	$\dot{\phi}_4$	u_{w_4}		
Self 8		δ_2	δ_3	δ_4		
Self 9		ϕ_2	ϕ_3	ϕ_4		
Self 10		$\dot{\phi}_2$	$\dot{\phi}_3$	$\dot{\phi}_4$		
Self 11		u_{w_2}	u_{w_3}	u_{w_4}		

The chosen selves are high-order selves, being either 3D or 4D. A graphical representation of the 3D selves is given in Figure 7.9, where the self (blue), failure data (black) and the ABs (red) corresponding to Self 11 are plotted in the same space (Figure 7.9a). Similar to the 2D case, the ABs are generated through a negative-selection process as a means to optimize and maximize the number of ABs within the detection space. In this case, the generated 3D antibodies for Self 11 are able to detect the failure data since its trajectory falls right inside the volume of specific individual ABs. In Figure 7.9b, a clearer depiction of the self and failure data is presented. Self 11 is a good projection of the hyperspace given that the self and faulty data are distinct, thus increasing the faulty agent detection rate.

The fault detection results are presented in Figure 7.10 for Self 1 and Self 2. Even though various ABs are sporadically activated, these detection results are representative and promising. The detection results confirm that the other agents in the network are capable of detecting a fault within a neighboring faulty agent. Even though only representative results for fault detection in Agent 1 are shown in this section, these selves should be completed with the selves reserved for fault detection in Agents 2, 3 & 4. Thereafter, a fault identification logic can be created to detect which agent is faulty based on the activation of predefined AB sets.



(a) 3D representation of the self (blue), failure data (black) and the ABs (red)



(b) 3D representation of the self (blue) and failure data (black)*Figure 7.9* 3D representation of Self 11 (per Table 7.5)




Figure 7.10 Fault detection results from Self 1 and Self 2. The results confirm that Agents 2, 3 & 4 in the network are able to detect a fault in Agent 1.

Summary: This chapter presented the expansion of the AIS paradigm to a global level and self/non-self generation algorithms for a MAS. This architecture uses features from across multiple agents with a goal to detect abnormal conditions such as disturbances or faults and to further identify the faulty agent. A self representation using 2D detectors is laid out and tested. The high-order selves are introduced to better scale the AB generation for a MAS with a high number of agents. It is shown that with proper AB selection, DRs increase and FAs diminish, thus increasing the probability of detecting eventual faults or disturbances. Global fault detection results are presented for a new topology. New GHM selves are also introduced. Even in this budding phase, results show promising capabilities for the agent fault detection at the global level.

8. Discussion and Conclusion

In this dissertation, the capabilities of biologically inspired mechanisms, such as those found in the immune system, were investigated to solve MAS consensus and overall mission protection problems. A general architecture inspired by the functioning of these biological mechanisms, was designed to increase resilience of MAS missions operating under nominal or exogenous disturbances.

In review, the impact and significance of this research targets mission applications that involve the deployment of groups of autonomous vehicles with decentralized swarming capabilities and require advanced and novel technologies to increase overall mission performance while operating in complex and changing environments. It was discussed that MAS consensus algorithms have drawn much attention mainly for multiple uses in aerospace applications, namely spacecraft formation flying, sensor networks and unmanned air vehicle formations. The study reminded that these systems can be robots, vehicles, sensors, or process plants that work cooperatively to achieve certain specific tasks.

The advantage of using this bio-inspired approach was presented as having a distributed self-adaptive system that allows fast response to hostile invasions. Thus, compared to the single-agent, the MAS provides larger redundancy, higher robustness, and greater fault-tolerance. The overall aim is for the MAS to function autonomously and bypass collaboration with humans for repetitive, risky, and often critical missions as the distributed systems provide scalability and robustness during missions.

In continuation with previous studies at the ADCL on resilient control of aerospace systems, this work presented the design and implementation of a DMRAC using an immune-inspired adaptation law in order to mitigate the effects of known bounded disturbances on agents within a network under a directed communication topology.

With the main goal of getting the agents to operate within a distributed consensus protocol while minimizing the effect of disturbances, the work progressed as follows. Feedback linearization was used to modify the high-order nonlinear quadrotor model into four linear subsystems with no coupling. The chosen consensus algorithm consisted of a local feedback controller and interactions from the neighbors under fixed directed topologies. The effectiveness of the proposed architecture was examined through numerical simulations of four scenarios: consensus and tracking, consensus and formation, communication link attack and zero-dynamics attack (ZDA). These test cases demonstrated and evaluated the techniques at different levels by combining self-detection and diagnosis as a means to increase the safety of MAS missions, optimize endurance, and maintain performance even within hazardous operating environments.

Simulation results were presented in which the DMRAC was applied to a MAS of quadrotors with linearized dynamics while rejecting the disturbances given above. The results demonstrated that the proposed approach can effectively mitigate the effects of exogenous bounded disturbances and communication link attacks in directed graph topologies and, in the majority of scenarios, the DMRAC outperforms the baseline controller. In the case of the ZDAs, it was observed that the DMRAC was not able to compensate due to the innate stealthiness and unbounded nature of the attack. The stability proof of the approach was carried out using a time-varying Lyapunov function. The proposed controller was proven to be stable in the presence of external time-varying bounded disturbances. It is acknowledged that the current work is based on simulation and results for the same tests with physical systems might be different.

Finally, a global health management (GHM) architecture was designed with the goal to detect faults and disturbances within the network. This is shown by distinctly selecting GHM selves. The selected initial cases show that with proper AB selection, DRs increase and FAs diminish, thus increasing the probability of detecting eventual faults or disturbances. If the faults can be detected by an agent and/or its neighbors, then the system is robust. Results show that ACs in an agent can be detected by its neighbors. While fault detection is achieved, fault identification is a future work.

8.1. Future Work on Distributed MRAC

Future work on the application of the proposed DMRAC following this study can include the use of the generalizing assumption that some states are not measurable and thereby set up a state observer (Al Janaideh et al., 2019). Another reasonable assumption could be that disturbances are unknown and would need to be estimated before being rejected by the adaptive controller. In addition, the work can further be extended to include case studies with unknown modeling parameters in the quadrotors' dynamics. Other possible research continuations can pursue generalizing or extending the proposed results to other agent dynamics, possibly spacecrafts or fixed-winged aircraft.

8.2. Future Work on Global Health Monitoring

To further the groundwork on the GHM, more tests can be carried out on agents by injecting different ACs (hardware failures and cyber-attacks) in diverse topologies. The parameters used during the antibody generation algorithms can be varied with a view to better assess their impact in the GHM. This can then be followed by a comparative analysis with the same tests for the 2D selves. Additional tests can also be carried out to compare the high-order and 2D selves using similar features. An extension can be made for faulty agent identification and feature redefinition to detect topological changes. The quadrotor model could include unmodeled dynamics to the linearized model for increased modeling accuracy.

The development and implementation of an adaptive global health management could increase robustness of the abnormal condition architecture. In this study, improper ABs were manually detected and omitted within each self to increase the detection rate. In future work, this would ideally require an intelligent system to automatically find and exclude ill-defined ABs.

Another extension can include the output of the local AIS as part of the controller. This would lead to the creation of a hybrid AIS framework for the network using both reinforcement learning methods and algorithms. The network would, in time, develop its own set of antibodies capable of rejecting a variety of disturbances, thus establishing an acquired immune system.

REFERENCES

- Ajwad, S. A., Moulay, E., Defoort, M., Ménard, T., & Coirault, P. (2021). Leader-Following Consensus of Second-Order Multi-Agent Systems With Switching Topology and Partial Aperiodic Sampled Data. *IEEE Control* Systems Letters, 5, 1567-1572.
- Al Janaideh, M., Hammad, E., Farraj, A., & Kundur, D. (2019). Mitigating Attacks With Nonlinear Dynamics on Actuators in Cyber-Physical Mechatronic Systems. *IEEE Transactions on Industrial Informatics*, 15(9), 4845-4856.
- Altug, E., Ostrowski, J. P., & Taylor, C. J. (2005). Control of a Quadrotor Helicopter Using Dual Camera Visual Feedback. The International Journal of Robotics Research, 24(5), 329-341.
- Angeli, D., & Bliman, P.-A. (2006). Stability of leaderless discrete-time multi-agent systems. *Mathematics of Control, Signals, and Systems*, 18(4), Springer Science and Business Media LLC, 293–322.
- Arfat, Y., & Eassa, F. (2016). A Survey on Fault Tolerant Multi Agent System. International Journal of Information Technology and Computer Science, 8(9), 39-48.
- Aziz, A. S. A., Hanafi, S. E.-O., & Hassanien, A. E. (2014). Multi-agent Artificial Immune System for Network Intrusion Detection and Classification. In International Joint Conference SOCO'14-CISIS'14-ICEUTE'14 (pp. 145–154). Springer International Publishing.
- Azzawi, D. A., Moncayo, H., Perhinschi, M. G., Perez, A., & Togayev, A. (2016). Comparison of immunity-based schemes for aircraft failure detection and identification. *Engineering Applications of Artificial Intelligence*, 52, 181–193.
- Babazadeh, R., & Selmic, R. (2020). Distance-Based Multiagent Formation Control With Energy Constraints Using SDRE. *IEEE Transactions on Aerospace and Electronic Systems*, 56(1), 41-56.
- Bai, J., Wen, G., Rahmani, A., & Yu, Y. (2018). Consensus for the fractional-order double-integrator multi-agent systems based on the sliding mode estimator. *IET Control Theory & Applications*, 12(5), 621-628.
- Balas, M. J., & Frost, S. A. (2014). Robust Adaptive Model Tracking for Distributed Parameter Control of Linear Infinite-dimensional Systems in Hilbert Space. *IEEE/CAA Journal of Automatica Sinica*, 1(3), 294-301.

- Beard, R., Cao, C., & Hovakimyan, N. (2006). An L₁ Adaptive Pitch Controller for Miniature Air Vehicles. In AIAA Guidance, Navigation, and Control Conference and Exhibit. American Institute of Aeronautics and Astronautics.
- Blanke, M., Kinnaert, M., Lunze, J., & Staroswiecki, M. (2016). Diagnosis and Fault-Tolerant Control. Springer Berlin Heidelberg.
- Bobtsov, A., & Pyrkin, A. (2007). A new approach to MRAC problem with disturbance rejection. *IFAC Proceedings Volumes*, 40(13), 92-97. (9th IFAC Workshop on Adaptation and Learning in Control and Signal Processing)
- Bullo, F., Cortes, J., & Martinez, S. (2009). Distributed Control of Robotic Networks. Applied Mathematics Series, Princeton University Press, N.J.
- Cai, M., Xiang, Z., & Guo, J. (2016). Adaptive finite-time fault-tolerant consensus protocols for multiple mechanical systems. *Journal of the Franklin Institute*, 353(6), Elsevier BV, 1386–1408.
- Cai, N., Cao, J.-W., Ma, H.-Y., & Wang, C.-X. (2013). Swarm Stability Analysis of Nonlinear Dynamical Multi-Agent Systems via Relative Lyapunov Function. *Arabian Journal for Science and Engineering*, 39(3), Springer Science and Business Media LLC, 2427–2434.
- Campion, M., Ranganathan, P., & Faruque, S. (2018). A Review and Future Directions of UAV Swarm Communication Architectures. *IEEE International* Conference on Electro/Information Technology (EIT), 0903-0908.
- Cao, C., & Hovakimyan, N. (2007). Novel \mathcal{L}_1 Neural Network Adaptive Control Architecture With Guaranteed Transient Performance. *IEEE Transactions on Neural Networks*, 18(4), 1160–1171.
- Cao, Y., Yu, W., Ren, W., & Chen, G. (2013). An Overview of Recent Progress in the Study of Distributed Multi-Agent Coordination. *IEEE Transactions on Industrial Informatics*, 9(1), 427-438.
- Castro, L. N. D., & Zuben, F. J. V. (2002). The Clonal Selection Algorithm with Engineering Applications. In GECCO 2002 - Workshop Proceedings, 36–37.
- Chan, R., & Lau, H. (2012). An AIS-based optimal control framework for longevity and task achievement of multi-robot systems. Numerical Algebra, Control and Optimization, 2(1), 45–56.

- Chen, H., Wang, X., Shen, L., & Cong, Y. (2021). Formation flight of fixed-wing UAV swarms: A group-based hierarchical approach. *Chinese Journal of Aeronautics*, 34(2), 504–515.
- Chen, S., Ho, D. W. C., Li, L., & Liu, M. (2015). Fault-Tolerant Consensus of Multi-Agent System With Distributed Adaptive Protocol. *IEEE Transactions* on Cybernetics, 45(10), Institute of Electrical and Electronics Engineers, 2142–2155.
- Chen, W., Zhou, J., & Wei, H. (2006). Compensatory Controller Based on Artificial Immune System. *IEEE International Conference on Mechatronics and* Automation, 1608–1613.
- Cheng, K., & Dasgupta, P. (2007). Dynamic Area Coverage using Faulty Multi-Agent Swarms. 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'07), 17–23.
- Clotet, X., Moyano, J., & León, G. (2018). A real-time anomaly-based IDS for cyber-attack detection at the industrial process level of Critical Infrastructures. International Journal of Critical Infrastructure Protection, 23, 11–20.
- Cui, Y., Liu, Y., Zhang, W., & Alsaadi, F. E. (2021). Sampled-Based Consensus for Nonlinear Multiagent Systems With Deception Attacks: The Decoupled Method. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(1), 561–573.
- Das, A., Lewis, F., & Subbarao, K. (2009). Dynamic inversion with zero-dynamics stabilisation for quadrotor control. *IET Control Theory & Applications*, 3(3), Institution of Engineering and Technology, 303–314.
- Dasgupta, S., D. Forrest. (1999). Artificial immune systems in industrial applications. In Proceedings of the Second International Conference on Intelligent Processing and Manufacturing of Materials. IPMM'99, (Cat. No.99EX296). IEEE, pp. 257–267 Vol. 1.
- Davis, J., Perhinschi, M., & Moncayo, H. (2010). Evolutionary Algorithm for Artificial-Immune-System-Based Failure-Detector Generation and Optimization. Journal of Guidance Control and Dynamics, 33(2), 305–320.
- Davoodi, M., Meskin, N., & Khorasani, K. (2018). Integrated Fault Diagnosis and Control Design of Linear Complex Systems. Control Robotics and Sensors Series 121, Institution of Engineering and Technology, Stevenage, UK.

- De La Torre, G., & Yucelen, T. (2018). Adaptive architectures for resilient control of networked multiagent systems in the presence of misbehaving agents. *International Journal of Control*, 91(3), 495–507.
- Dimarogonas, D. V., & Johansson, K. H. (2008). On the stability of distance-based formation control. 47th IEEE Conference on Decision and Control, 1200–1205.
- Ding, L., Han, Q., Ge, X., & Zhang, X. (2018). An Overview of Recent Advances in Event-Triggered Consensus of Multiagent Systems. *IEEE Transactions on Cybernetics*, 48(4), 1110–1123.
- Ding, Z. (2014). Consensus control of a class of Lipschitz nonlinear systems. International Journal of Control, 87(11), 2372–2382.
- Dogan, K., Gruenwald, B. C., Yucelen, T., Muse, J. A., & Butcher, E. (2017). Distributed adaptive control of networked multiagent systems with heterogeneous actuator dynamics. 2017 American Control Conference (ACC).
- Dressler, F., & Akan, O. B. (2010). A survey on bio-inspired networking. Computer Networks, 54 (6), 881–900. (New Network Paradigms)
- Duan, H., Shao, S., Su, B., & Zhang, L. (2010). New development thoughts on the bio-inspired intelligence based control for unmanned combat aerial vehicle. *Science China Technological Sciences*, 53, 2025-2031.
- Dutta, R. (2016). Cooperative control of autonomous network topologies. *ProQuest Dissertations and Theses*, The University of Texas at San Antonio.
- Dutta, R., Sun, L., Kothari, M., Sharma, R., & Pack, D. (2014). A cooperative formation control strategy maintaining connectivity of a multi-agent system. *IEEE International Conference on Intelligent Robots and Systems*, 1189-1194.
- D'Souza, J. M., Guruprasad, K., & Padman, A. (2019). A realistic simulation platform for multi-quadcopter search using downward facing cameras. *Computers & Electrical Engineering*, 74, 184–195.
- F., D., & Moncayo, H. (2020). Hybrid Immunized Swarm Optimization Concept for Resilient Coordinated Missions. In AIAA SciTech 2020 Forum, Orlando FL, January 6-10.
- F., D., & Moncayo, H. (2021). Immunity-based Consensus Algorithm for Increased Resilience in Multiagent Missions. In AIAA SciTech 2021 Forum, Virtual Event.

- Fattahi, M., & Afshar, A. (2019). Fault tolerant consensus control of multi-agent systems with switching topology. SN Applied Sciences, 1(599).
- Fax, J., & Murray, R. (2004). Information Flow and Cooperative Control of Vehicle Formations. *IEEE Transactions on Automatic Control*, 49(9), 1465–1476.
- Feddema, J., Lewis, C., & Schoenwald, D. (2002). Decentralized control of cooperative robotic vehicles: Theory and application. *IEEE Transactions on Robotics and Automation*, 18, 852 – 864.
- Feng, Z., Hu, G., & Wen, G. (2015). Distributed consensus tracking for multi-agent systems under two types of attacks. *International Journal of Robust and Nonlinear Control*, 26(5), Wiley, 896–918.
- Feng, Z., Wen, G., & Hu, G. (2017). Distributed Secure Coordinated Control for Multiagent Systems Under Strategic Attacks. *IEEE Transactions on Cybernetics*, 47(5), 1273–1284.
- Ferrari, R. M. G., Parisini, T., & Polycarpou, M. M. (2012). Distributed Fault Detection and Isolation of Large-Scale Discrete-Time Nonlinear Systems: An Adaptive Approximation Approach. *IEEE Transactions on Automatic Control*, 57(2), 275–290.
- Forrest, S., Allen, L., Perelson, A., & Cherukuri, R. (1994). Self-nonself discrimination in a computer. Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy, 202–212.
- Gallehdari, Z., Meskin, N., & Khorasani, K. (2017). An H_{∞} cooperative fault recovery control of multi-agent systems. *Automatica*, 84, Elsevier BV, 101–108.
- Gao, Q., Huang, H., Feng, G., Liu, L., & Wang, Y. (2013). Robust H_{∞} Stabilization of Uncertain T-S Fuzzy Systems via Dynamic Integral Sliding Mode Control. *IFAC Proceedings Volumes*, 46(20), Elsevier BV, 485–490.
- Garcia, D., Moncayo, H., Perez, A., & Jain, C. (2016). Low cost implementation of a biomimetic approach for UAV health management. 2016 American Control Conference (ACC), Boston MA, July 6-8, 2265–2270.
- Gong, P. (2016). Distributed consensus of non-linear fractional-order multi-agent systems with directed topologies. *IET Control Theory & Applications*, 10(18), 2515–2525.

- Gong, P., Lan, W., & Han, Q.-L. (2020). Robust adaptive fault-tolerant consensus control for uncertain nonlinear fractional-order multi-agent systems with directed topologies. *Automatica*, 117, Elsevier BV, 109011.
- González, F., Dasgupta, D., & Niño, L. F. (2003). A Randomized Real-Valued Negative Selection Algorithm. In *Lecture notes in computer science* (pp. 261–272). Springer Berlin Heidelberg.
- Graaff, A., & Engelbrecht, A. (2006). Optimised Coverage of Non-self with Evolved Lymphocytes in an Artificial Immune System. International Journal of Computational Intelligence Research Research India Publications, 2(2), 973–1873.
- Gulzar, M. M., Rizvi, S. T. H., Javed, M. Y., Munir, U., & Asif, H. (2018). Multi-Agent Cooperative Control Consensus: A Comparative Review. *Electronics*, 7(2).
- Guo, S., Li, Z., Niu, Y., & Wu, L. (2020). Consensus disturbance rejection control of directed multi-agent networks with extended state observer. *Chinese Journal of Aeronautics*, 33(5), 1486–1493.
- Guzey, H. M. (2017). Adaptive consensus-based formation control of fixed-wing MUAV's., 4th International Conference Actual Problems of Unmanned Aerial Vehicles Developments, 184–187.
- Han, Z., Lin, Z., Fu, M., & Chen, Z. (2015). Distributed coordination in multi-agent systems: a graph Laplacian perspective. Frontiers of Information Technology & Electronic Engineering, 16(6), Zhejiang University Press, 429–448.
- He, W., Xu, B., Han, Q., & Qian, F. (2020). Adaptive Consensus Control of Linear Multiagent Systems With Dynamic Event-Triggered Strategies. *IEEE Transactions on Cybernetics*, 50(7), 2996–3008.
- Horn, R., & Johnson, C. (1990). Topics in Matrix Analysis. Cambridge University Press, Cambridge.
- Hovakimyan, N., & Cao, C. (2010). \mathcal{L}_1 Adaptive Control Theory. Society for Industrial and Applied Mathematics, Philadelphia.
- Huang, Z. (2017). Consensus Control of Multiple-Quadcopter Systems under Communication Delays. *Thesis*, Dalhousie University.

- Huang, Z., Pan, Y.-J., & Bauer, R. (2019). Leader–Follower Consensus Control of Multiple Quadcopters Under Communication Delays. *Journal of Dynamic* Systems, Measurement, and Control, 141(10).
- Humza, R., Scholz, O., Mokhtar, M., Timmis, J., & Tyrrell, A. (2009). Towards Energy Homeostasis in an Autonomous Self-Reconfigurable Modular Robotic Organism. In *Computation world: Future computing, service computation, cognitive, adaptive, content, patterns* (pp. 21–26).
- Isidori, A. (2013). The zero dynamics of a nonlinear system: From the origin to the latest progresses of a long successful story. *European Journal of Control*, 19(5), 369-378. (The Path of Control)
- Ismail, A. R., & Timmis, J. (2010). Towards Self-Healing Swarm Robotic Systems Inspired by Granuloma Formation. 15th IEEE International Conference on Engineering of Complex Computer Systems, 313–314.
- Ji, M., & Egerstedt, M. (2007). A Graph-Theoretic Characterization of Controllability for Multi-agent Systems. American Control Conference, 4588–4593.
- Ji, Z., & Dasgupta, D. (2004). Real-Valued Negative Selection Algorithm with Variable-Sized Detectors. In *Genetic and evolutionary computation – GECCO* 2004 (pp. 287–298). Springer Berlin Heidelberg.
- Ji, Z., & Dasgupta, D. (2009). V-detector: An efficient negative selection algorithm with "probably adequate" detector coverage. *Information Sciences*, 179(10), 1390–1406.
- Joo, Y., Qu, Z., & Namerikawa, T. (2020). Resilient Control of Cyber-Physical System Using Nonlinear Encoding Signal Against System Integrity Attacks. *IEEE Transactions on Automatic Control.*
- Kaneshige, J., & Krishnakumar, K. (2007). Artificial Immune System Approach for Air Combat Maneuvering. In Intelligent Computing: Theory and Applications V (Vol. 6560).
- Kang, S., Choi, H., & Kim, Y. (2013). Formation Flight and Collision Avoidance for Multiple UAVs using Concept of Elastic Weighting Factor. International Journal of Aeronautical and Space Sciences, 14(1), 75–84.
- Karr, C., Nishita, K., & Graham, K. (2005). Adaptive Aircraft Flight Control Simulation Based on an Artificial Immune System. Applied Intelligence, 23, 295–308.

- Kaufman, H., Barkana, I., & Sobel, K. (1997). Direct Adaptive Control Algorithms: Theory and Applications. Communications and Control Engineering, Springer, New York.
- Keliris, C., Polycarpou, M. M., & Parisini, T. (2013). A Distributed Fault Detection Filtering Approach for a Class of Interconnected Continuous-Time Nonlinear Systems. *IEEE Transactions on Automatic Control*, 58(8), Institute of Electrical and Electronics Engineers, 2032–2047.
- Keviczky, T., Borrelli, F., Fregene, K., Godbole, D., & Balas, G. J. (2008). Decentralized Receding Horizon Control and Coordination of Autonomous Vehicle Formations. *IEEE Transactions on Control Systems Technology*, 16(1), 19–33.
- Khalil, H. (2002). Nonlinear Systems. Prentice Hall, NJ. Pearson Education.
- Khalili, M., Zhang, X., Polycarpou, M. M., Parisini, T., & Cao, Y. (2018). Distributed adaptive fault-tolerant control of uncertain multi-agent systems. *Automatica*, 87, Elsevier BV, 142–151.
- Kim, J., Back, J., Park, G., Lee, C., Shim, H., & Voulgaris, P. G. (2020). Neutralizing zero dynamics attack on sampled-data systems via generalized holds. *Automatica*, 113.
- Kim, J., & Shim, H. (2019). A countermeasure against zero-dynamics sensor attack via generalized hold feedback. In 2019 58th annual conference of the society of instrument and control engineers of japan (SICE). IEEE.
- Kolaric, P., Chen, C., Dalal, A., & Lewis, F. L. (2018). Consensus controller for multi-UAV navigation. *Control Theory and Technology*, 16(2), Springer Science and Business Media LLC, 110–121.
- Krstic, M., Kanellakopoulos, I., & Kokotovic, P. V. (1995). Nonlinear and adaptive control design. Wiley series in adaptive and learning systems for signal processing, communication and control. Wiley, New York, NY.
- Kumar, K. K. (2003). Artificial Immune System Approaches for Aerospace Applications. AIAA 41st Aerospace Sciences Meeting and Exhibit.
- Kuriki, Y., & Namerikawa, T. (2013). Formation control of UAVs with a fourth-order flight dynamics. 52nd IEEE Conference on Decision and Control, 6706–6711.

- Lafferriere, G., Williams, A., Caughman, J., & Veerman, J. (2005). Decentralized control of vehicle formations. Systems & Control Letters, 54, 899–910.
- Lau, H., Bate, I., Cairns, P., & Timmis, J. (2011). Adaptive data-driven error detection in swarm robotics with statistical classifiers. *Robotics and Autonomous Systems*, 59(12), Elsevier BV, 1021–1035.
- Lau, H., & Wong, V. (2006). An immunity-based distributed multiagent-control framework. *IEEE Transactions on Systems, Man, and Cybernetics - Part A:* Systems and Humans, 36(1), 91–108.
- Lau, H. Y., & Qiu, X. (2014). An Artificial Immune Systems (AIS)-based Unified Framework for General Job Shop Scheduling. *IFAC Proceedings Volumes*, 47(3), 6186–6191. (19th IFAC World Congress)
- Lewis, F. L., Zhang, H., Hengster-Movric, K., & Das, A. (2014). Cooperative Control of Multi-Agent Systems. Springer London.
- Li, M., & Deng, F. (2021). Cluster consensus of nonlinear multi-agent systems with Markovian switching topologies and communication noises. *ISA Transactions*, 116, 113–120.
- Li, X.-M., Zhou, Q., Li, P., Li, H., & Lu, R. (2020). Event-Triggered Consensus Control for Multi-Agent Systems Against False Data-Injection Attacks. *IEEE Transactions on Cybernetics*, 50(5), 1856–1866.
- Li, Y.-X., & Yang, G.-H. (2019). Graph-Theory-Based Decentralized Adaptive Output-Feedback Control for a Class of Nonlinear Interconnected Systems. *IEEE Transactions on Cybernetics*, 49(7), 2444–2453.
- Li, Z., Duan, Z., & Lewis, F. L. (2014). Distributed robust consensus control of multi-agent systems with heterogeneous matching uncertainties. *Automatica*, 50(3), 883–889.
- Li, Z., Ren, W., Liu, X., & Fu, M. (2013). Consensus of Multi-Agent Systems with General Linear and Lipschitz Nonlinear Dynamics Using Distributed Adaptive Protocols. *IEEE Transactions on Automatic Control*, 58(7), 1786-1791.
- Lin, Z., Broucke, M., & Francis, B. (2004). Local Control Strategies for Groups of Mobile Autonomous Agents. *IEEE Transactions on Automatic Control*, 49(4), 622–629.

- Liu, S., Wei, G., Song, Y., & Liu, Y. (2016). Extended Kalman filtering for stochastic nonlinear systems with randomly occurring cyber attacks. *Neurocomputing*, 207, 708–716.
- Liu, Y., & Jia, Y. (2012). Adaptive leader-following consensus control of multi-agent systems using model reference adaptive control approach. *IET Control Theory* & Applications, 6(13), 2002–2008.
- Liu, Z., & Wong, W. S. (2013). Choice-based cluster consensus in multi-agent systems. In *Proceedings of the 32nd chinese control conference* (pp. 7285–7290).
- Lu, J., & Chen, G. (2005). A time-varying complex dynamical network model and its controlled synchronization criteria. *IEEE Transactions on Automatic Control*, 50(6), 841-846.
- Lui, D. G., Petrillo, A., & Santini, S. (2021). Distributed model reference adaptive containment control of heterogeneous multi-agent systems with unknown uncertainties and directed topologies. *Journal of the Franklin Institute*, 358(1), 737-756.
- Luo, J. (2013). Adaptive Control for Distributed Multi-Agent Coordination. Dissertation, University of Connecticut.
- Mange, J. (2013). Artificial immune systems and particle swarm optimization for solutions to the general adversarial agents problem. *Dissertation*, West Virginia University.
- Mao, Y., Akyol, E., & Zhang, Z. (2019). A Novel Defense Strategy Against Zero-Dynamics Attacks in Multi-Agent Systems. *IEEE 58th Conference on Decision and Control (CDC)*, 3563–3568.
- Mao, Y., Jafarnejadsani, H., Zhao, P., Akyol, E., & Hovakimyan, N. (2019). Detectability of Intermittent Zero-Dynamics Attack in Networked Control Systems. *IEEE 58th Conference on Decision and Control 2019*, 5605–5610.
- Mao, Y., Jafarnejadsani, H., Zhao, P., Akyol, E., & Hovakimyan, N. (2020). Novel Stealthy Attack and Defense Strategies for Networked Control Systems. *IEEE Transactions on Automatic Control*, 65, 3847–3862.
- Mehdifar, F., Bechlioulis, C. P., Hashemzadeh, F., & Baradarannia, M. (2020). Prescribed Performance Distance-Based Formation Control of Multi-Agent Systems (Extended Version). Automatica, 119.

- Mei, J., Ren, W., & Chen, J. (2016). Distributed Consensus of Second-Order Multi-Agent Systems With Heterogeneous Unknown Inertias and Control Gains Under a Directed Graph. *IEEE Transactions on Automatic Control*, 61(8), 2019–2034.
- Meng, Z., Cao, Y., & Ren, W. (2010). Stability and convergence analysis of multi-agent consensus with information reuse. *International Journal of Control*, 83(5), 1081–1092.
- Mesbahi, M., & Egerstedt, M. (2010). *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, Princeton NJ.
- Mohamed, M. E., Samir, B. B., & Abdullah, A. (2010). Immune multiagent system for network intrusion detection using non-linear classification algorithm. *International Journal of Computer Applications*, 12(7), Foundation of Computer Science, 7–12.
- Moncayo, H., Perhinschi, M., Wilburn, B., Wilburn, J., & Karas, O. (2012). UAV Adaptive Control Laws Using Non-Linear Dynamic Inversion Augmented with an Immunity-based Mechanism. AIAA Guidance, Navigation, and Control Conference, Minneapolis MN, August 13-16.
- Moncayo, H., Perhinschi, M. G., & Davis, J. (2011). Artificial-Immune-System-Based Aircraft Failure Evaluation over Extended Flight Envelope. Journal of Guidance, Control, and Dynamics, 34 (4), 989–1001.
- Morse, A. (1980). Global stability of parameter-adaptive control systems. *IEEE Transactions on Automatic Control*, 25(3), 433–439.
- Naghnaeian, M., Hirzallah, N., & Voulgaris, P. G. (2015). Dual rate control for security in cyber-physical systems. In 2015 54th IEEE Conference on Decision and Control (CDC) (pp. 1415–1420).
- Nguyen, N. (2018). Model Reference Adaptive Control a Primer. Springer International Publishing, Switzerland.
- Oh, K.-K., & Ahn, H.-S. (2011). Formation control of mobile agents based on inter-agent distance dynamics. Automatica, 47(10), 2306–2312.
- Oh, K.-K., Park, M.-C., & Ahn, H.-S. (2015). A survey of multi-agent formation control. *Automatica*, 53, 424–440.

- O'Keeffe, J., Tarapore, D., Millard, A., & Timmis, J. (2017). Towards fault diagnosis in robot swarms: An online behaviour characterisation approach. 18th Annual Conference on Towards Autonomous Robotic Systems, TAROS 2017, 10454 LNAI, 393–407.
- O'Keeffe, J., Tarapore, D., Millard, A. G., & Timmis, J. (2018). Adaptive Online Fault Diagnosis in Autonomous Robot Swarms. Frontiers in Robotics and AI, 5, 131.
- Olfati-Saber, R., & Murray, R. M. (2004). Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9), 1520–1533.
- Olfati-Saber, R., Fax, J. A., & Murray, R. (2007). Consensus and Cooperation in Networked Multi-Agent Systems. Proceedings of the IEEE, 95, 215–233.
- Pang, Y., Xia, H., & Grimble, M. J. (2020). Resilient Nonlinear Control for Attacked Cyber-Physical Systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(6), 2129–2138.
- Parker, L. (1993). Designing control laws for cooperative agent teams. In [1993] Proceedings International Conference on Robotics and Automation. IEEE.
- Pasqualetti, F., Dörfler, F., & Bullo, F. (2013). Attack Detection and Identification in Cyber-Physical Systems. *IEEE Transactions on Automatic Control*, 58.
- Pasqualetti, F., Dörfler, F., & Bullo, F. (2012a). Attack Detection and Identification in Cyber-Physical Systems – Part II: Centralized and Distributed Monitor Design. Part 2 Math.OC.
- Pasqualetti, F., Dörfler, F., & Bullo, F. (2012b). Attack Detection and Identification in Cyber-Physical Systems – Part I: Models and Fundamental Limitations. *Part* 1 Math.OC.
- Peng, Z., Wang, D., Zhang, H., Sun, G., & Wang, H. (2013). Distributed model reference adaptive control for cooperative tracking of uncertain dynamical multi-agent systems. *IET Control Theory & Applications*, 7(8), Institution of Engineering and Technology, 1079–1087.
- Perez, A. E., Moncayo, H., Perhinschi, M., Al Azzawi, D., & Togayev, A. (2015). A Bio-Inspired Adaptive Control Compensation System for an Aircraft Outside Bounds of Nominal Design. *Journal of Dynamic Systems, Measurement, and Control*, 137(9).

- Perhinschi, M. G., & Moncayo, H. (2018). Artificial Immune System for Comprehensive and Integrated Aircraft Abnormal Conditions Management. In J. Valasek (Ed.), Advances in Computational Intelligence and Autonomy for Aerospace Systems (pp. 147–218).
- Perhinschi, M. G., Moncayo, H., & Al Azzawi, D. (2014). Integrated Immunity-Based Framework for Aircraft Abnormal Conditions Management. *Journal of Aircraft*, 51(6), 1726–1739.
- Perhinschi, M. G., Moncayo, H., & Azzawi, D. A. (2013). Development of Immunity-Based Framework for Aircraft Abnormal Conditions Detection, Identification, Evaluation, and Accommodation. AIAA GNC Conference.
- Podlubny, I. (1998). Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of their Solution and some of their Applications. Academic Press, San Diego.
- Pourseif, T., Taheri Andani, M., Ramezani, Z., & Pourgholi, M. (2017). Model Reference Adaptive Control for Robot Tracking Problem: Design & Performance Analysis. *International Journal of Control Science and Engineering*, 7(1), 18–23.
- Qin, J., Qichao, M., Shi, Y., & Wang, L. (2017). Recent Advances in Consensus of Multi-Agent Systems: A Brief Survey. *IEEE Transactions on Industrial Electronics*, 64(6), 4972–4983.
- Rahmani, A., Ji, M., Mesbahi, M., & Egerstedt, M. (2009). Controllability of Multi-Agent Systems from a Graph-Theoretic Perspective. SIAM Journal on Control and Optimization, 48(1), 162–186.
- Ren, W. (2007). Consensus strategies for cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 1(2), 505–512.
- Ren, W., & Beard, R. W. (2008). Distributed Consensus in Multi-vehicle Cooperative Control. Springer-Verlag, London.
- Ren, W., Moore, K., & Chen, Y. (2006). High-Order Consensus Algorithms in Cooperative Vehicle Systems. In 2006 IEEE International Conference on Networking, Sensing and Control (pp. 457–462). IEEE.
- Ren, W., Moore, K., & Chen, Y. (2007). High-Order and Model Reference Consensus Algorithms in Cooperative Control of Multi-Vehicle Systems. *Journal of Dynamic Systems Measurement and Control – ASME*, 129, 678–688.

- Rocha, A. D., Lima-Monteiro, P., Parreira-Rocha, M., & Barata, J. (2017). Artificial immune systems based multi-agent architecture to perform distributed diagnosis. *Journal of Intelligent Manufacturing*, 30(4), 2025–2037.
- Rocha, A. E. (2016). Development of Fault Tolerant Adaptive Control Laws for Aerospace Systems. *ProQuest Dissertations and Theses*, Embry-Riddle Aeronautical University.
- Rohrs, C., Valavani, L., Athans, M., & Stein, G. (1985). Robustness of continuous-time adaptive control algorithms in the presence of unmodeled dynamics. *IEEE Transactions on Automatic Control*, 30(9), 881–889.
- Salehi, S., & Selamat, A. (2011). Hybrid simple artificial immune system (SAIS) and particle swarm optimization (PSO) for spam detection. 2011 Malaysian Conference in Software Engineering, 124–129.
- Seresht, N. A., & Azmi, R. (2014). MAIS-IDS: A distributed intrusion detection system using multi-agent AIS approach. Engineering Applications of Artificial Intelligence, 35, 286–298.
- Shames, I., Teixeira, A. M., Sandberg, H., & Johansson, K. H. (2011). Distributed fault detection for interconnected second-order systems. *Automatica*, 47(12), 2757–2764.
- Shamma, J. S. (Ed.). (2007). Cooperative Control of Distributed Multi-Agent Systems. John Wiley & Sons Ltd, Chichester, West Sussex PO19 8SQ, England.
- Shi, G., & Hong, Y. (2009). Distributed coordination of multi-agent systems with switching structure and input saturation. Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, 895–900.
- Shoham, Y., & Leyton-Brown, K. (2009). Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. Cambridge University Press, Cambridge.
- Singh Chingtham, T., Sahoo, G., & Ghose, M. K. (2011). An Artificial Immune System Model for Multi-Agents Resource Sharing in Distributed Environments. International Journal on Computer Science and Engineering, 2(5), 1813–1818.
- Skogestad, S., & Postlethwaite, I. (2005). Multivariable Feedback Control: Analysis and Design. In (Vols. 2 Norwegian University of Science and Technology, University of Leicester, John Wiley & Sons Ltd, England).

- Stirling, T., Roberts, J., Zufferey, J., & Floreano, D. (2012). Indoor navigation with a swarm of flying robots. 2012 IEEE International Conference on Robotics and Automation, 4641–4647.
- Sun, J., Geng, Z., Lv, Y., Li, Z., & Ding, Z. (2016). Distributed Adaptive Consensus Disturbance Rejection for Multi-Agent Systems on Directed Graphs. *IEEE Transactions on Control of Network Systems*, 5(1), 629–639.
- Sun, X., & Cassandras, C. G. (2016). Optimal dynamic formation control of multi-agent systems in constrained environments. Automatica, 73, 169–179.
- Taheri, M., Khorasani, K., Shames, I., & Meskin, N. (2020). Undetectable Cyber Attacks on Communication Links in Multi-Agent Cyber-Physical Systems. 59th IEEE Conference on Decision and Control, Systems and Control, 3764–3771.
- Tahir, A., Böling, J., Haghbayan, M.-H., Toivonen, H. T., & Plosila, J. (2019). Swarms of Unmanned Aerial Vehicles — A Survey. Journal of Industrial Information Integration, 16.
- Tahoun, A., & Arafa, M. (2021). Cooperative control for cyber–physical multi-agent networked control systems with unknown false data-injection and replay cyber-attacks. *ISA Transactions*, 110, 1–14.
- Takahashi, K., & Yamada, T. (1998). Application of an Immune Feedback Mechanism to Control Systems. JSME International Journal Series C, 41(2), 184–191.
- Tarao, M., & Okamoto, T. (2016). Toward an Artificial Immune Server against Cyber Attacks: Enhancement of Protection against DoS Attacks. *Proceedia Computer* Science, 96, 1137–1146.
- Teixeira, A., Shames, I., Sandberg, H., & Johansson, K. H. (2015). A secure control framework for resource-limited adversaries. Automatica, 51, 135–148.
- Timmis, J., Andrews, P., & Hart, E. (2010). On artificial immune systems and swarm intelligence. Swarm Intelligence, 4(4), 247–273.
- Timmis, J., Ismail, A., Bjerknes, J., & Winfield, A. (2016). An immune-inspired swarm aggregation algorithm for self-healing swarm robotic systems. *Biosystems*, 146, 60–76. (Information Processing in Cells and Tissues)
- Vu, D. V., Trinh, M. H., & Ahn, H.-S. (2020). Distance-Based Formation Tracking with Unknown Bounded Reference Velocities. *ICCAS*, *IEEE*.

- Wang, J., Wang, C., Xin, M., Ding, Z., & Shan, J. (2020). Cooperative Control of Multi-Agent Systems: An Optimal and Robust Perspective. A volume in Emerging Methodologies and Applications in Modelling, Elsevier Inc, Maryland.
- Wang, W., Long, J., Wen, C., & Huang, J. (2020). Recent advances in distributed adaptive consensus control of uncertain nonlinear multi-agent systems. *Journal* of Control and Decision, 7, 44–63.
- Wang, X., & Yang, G.-H. (2016). Distributed fault-tolerant control for a class of cooperative uncertain systems with actuator failures and switching topologies. *Information Sciences*, 370-371, 650–666.
- Wang, Y., Wu, Q., & Wang, Y. (2013). Distributed consensus protocols for coordinated control of multiple quadrotors under a directed topology. *IET Control Theory and Applications*, 7, 1780–1792.
- Wang, Y., Wu, Q., Wang, Y., & Yu, D. (2013). Consensus algorithm for multiple quadrotor systems under fixed and switching topologies. *Journal of Systems Engineering and Electronics*, 24(5), 818–827.
- Weng, L., Liu, Q., Xia, M., & Song, Y. (2014). Immune network-based swarm intelligence and its application to unmanned aerial vehicle (UAV) swarm coordination. *Neurocomputing*, 125, 134–141.
- Xue, D., Yao, J., Wang, J., Guo, Y., & Han, X. (2013). Formation control of multi-agent systems with stochastic switching topology and time-varying communication delays. *IET Control Theory & Applications*, 7(13), 1689–1698.
- Xue, R., & Cai, G. (2016). Formation Flight Control of Multi-UAV System with Communication Constraints. Journal of Aerospace Technology and Management, 8(2), 203–210.
- Yang, H., Han, Q., Ge, X., Ding, L., Xu, Y., Jiang, B., & Zhou, D. (2020). Fault-Tolerant Cooperative Control of Multiagent Systems: A Survey of Trends and Methodologies. *IEEE Transactions on Industrial Informatics*, 16(1), 4–17.
- Yang, H., Li, T., Hu, X., Wang, F., & Zou, Y. (2014). A Survey of Artificial Immune System Based Intrusion Detection. The Scientific World Journal, 2014, 1–11.
- Yang, H., Staroswiecki, M., Jiang, B., & Liu, J. (2011). Fault tolerant cooperative control for a class of nonlinear multi-agent systems. Systems & Control Letters, 60(4), 271–277.

- Yang, P., Xu, M., Li, D., Liu, Z., & Huang, Y. (2019). Distributed fault tolerant consensus control for multi-agent system with actuator fault based on adaptive observer. *Transactions of the Institute of Measurement and Control*, 41(15), SAGE Publications, 4207–4217.
- Yang, Z., Zhang, Q., & Chen, Z. (2019). Formation Control of Multi-Agent Systems with Region Constraint. Complexity, 2019, 1-6.
- Yoo, S. J. (2013). Fault-tolerant control of strict-feedback non-linear time-delay systems with prescribed performance. *IET Control Theory & Applications*, 7, 1553–1561.
- Yu, W., Chen, G., Cao, M., & Kurths, J. (2010). Second-Order Consensus for Multiagent Systems With Directed Topologies and Nonlinear Dynamics. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 40(3), 881–891.
- Yu, W., Ren, W., Zheng, W. X., Chen, G., & Lü, J. (2013). Distributed control gains design for consensus in multi-agent systems with second-order nonlinear dynamics. *Automatica*, 49(7), 2107–2115.
- Yuan, C., Zeng, W., & Dai, S.-L. (2019). Distributed model reference adaptive containment control of heterogeneous uncertain multi-agent systems. *ISA Transactions*, 86, 73–86.
- Yucelen, T., & Egerstedt, M. (2012). Control of multiagent systems under persistent disturbances. 2012 American Control Conference (ACC), Montreal, June 27-29, 5264–5269.
- Zavlanos, M., & Pappas, G. (2005). Controlling Connectivity of Dynamic Graphs. Proceedings of the 44th IEEE Conference on Decision and Control, Sevilla, December 12-15, 6388–6393.
- Zhang, H., & Lewis, F. L. (2012). Adaptive cooperative tracking control of higher-order nonlinear systems with unknown dynamics. *Automatica*, 48(7), 1432–1439.
- Zhang, P., Xue, H., & Gao, S. (2019). Multi-Agent Fault-Tolerant Control Based on Distributed Adaptive Consensus. *IEEE Access*, 7, 135882–135895.
- Zhen, Z., Tao, G., Xu, Y., & Song, G. (2019). Multivariable adaptive control based consensus flight control system for UAVs formation. Aerospace Science and Technology, 93, 1270–9638.

- Zheng, D., Zhang, H., Zhang, J. A., & Wang, G. (2018). Consensus of multi-agent systems with faults and mismatches under switched topologies using a delta operator method. *Neurocomputing*, 315, 198–209.
- Zhou, J., Lu, J.-a., & Lu, J. (2006). Adaptive synchronization of an uncertain complex dynamical network. *IEEE Transactions on Automatic Control*, 51(4), 652–656.
- Zhu, Q. M., Wang, J., Wang, C., Xin, M., Ding, Z., & Shan, J. (2020). Cooperative Control of Multi-Agent Systems. Elsevier. (Series: Emerging Methodologies and Applications in Modelling)
- Zhu, W., Chen, B., & Yang, J. (2017). Consensus of fractional-order multi-agent systems with input time delay. *Fractional Calculus and Applied Analysis*, 20(1), 52–70.
- Zou, Y., Guan, M., & Wen, C. (2019). Distributed Adaptive Control for Distance-based Formation and Flocking control of Multi-Agent Systems. *IET Control Theory & Applications*, 13.
- Zuo, Z., Cao, X., & Wang, Y. (2020). Security control of multi-agent systems under false data injection attacks. *Neurocomputing*, 404, 240–246.
- Zuo, Z., Han, Q., Ning, B., Ge, X., & Zhang, X. (2018). An Overview of Recent Advances in Fixed-Time Cooperative Control of Multiagent Systems. *IEEE Transactions on Industrial Informatics*, 14(6), 2322–2334.

APPENDIX A: Global fault detection

Table 8.1

Detection rate and false alarms after disturbance injection in an agent

Disturbance in	agent I Detector set 1	Detector set 2	Detector set 3	Detector set 4
Detection Rate	64 04%	63 77%	51 55%	97 90%
False Alarms	38 25%	31.44%	27.21%	96.20%
raise marms	00.2070	01.4470	21.2170	30.2070
	Detector set 5	Detector set 6	Detector set 7	Detector set 8
Detection Rate	56.81%	58.81%	64.57%	54.95%
False Alarms	20.37%	34.61%	29.91%	43.18%
Disturbance in agent 2				
	Detector set 1	Detector set 2	Detector set 3	Detector set 4
Detection Rate	72.46%	97.77%	92.77%	97.94%
False Alarms	19.24%	86.96%	86.96%	86.93%
	Detector set 5	Detector set 6	Detector set 7	Detector set 8
Detection Rate	99.37%	96.77%	74.59%	69.93%
False Alarms	96.87%	100.00%	26.28%	38.81%
Disturbance in	agent 3			
	Detector set 1	Detector set 2	Detector set 3	Detector set 4
Detection Rate	0.00%	0.00%	53.58%	53.61%
False Alarms	0.00%	0.00%	100.00%	100.00%
	Detector set 5	Detector set 6	Detector set 7	Detector set 8
Detection Rate	53.61%	0.00%	53.61%	53.61%
False Alarms	100.00%	0.00%	100.00%	100.00%
Disturbance in agent 4				
	Detector set 1	Detector set 2	Detector set 3	Detector set 4
Detection Rate	73.69%	61.07%	62.23%	60.67%
False Alarms	39.78%	31.04%	36.61%	28.94%
	Detector set 5	Detector set 6	Detector set 7	Detector set 8
Detection Rate	54.18%	59.54%	91.44%	96.34%
False Alarms	51.05%	33.51%	82.19%	100.00%



Disturbance in agent 1

(a)



Disturbance in agent 2



Disturbance in agent 3

(c)



Disturbance in agent 4

(d)

Figure 8.1 Detection rate and false alarms after disturbance injection in an agent. The ideal detection case is for a detector to have the highest detection rate possible while having the least false alarms. Some graphs show high FA rates. The ABs generating FAs can be disregarded.

APPENDIX B: Publications

The following papers were prepared during the course of the research and published

in conference proceedings:

- F., D., and Moncayo, H., "Immunity-based Consensus Algorithm for Increased Resilience in Multiagent Missions," AIAA SciTech 2021 Forum, 2021.
- 2. F., D., and Moncayo, H., "Hybrid Immunized Swarm Optimization Concept for Resilient Coordinated Missions," AIAA SciTech 2020 Forum, 2020.
- 3. Rivera Lopez, K. P., Moncayo, H., Verberne, J. A., and F, D., "Design and Implementation of Intelligent Decision-Making Algorithms for Unmanned Aerial Vehicles Mission Protection", AIAA Scitech 2019 Forum.
- 4. Nshuti, C. , Moncayo, H., Engblom, W., and F, D., "Modeling, Simulation and Flight Testing to Support Proof of a Stratospheric Dual Aircraft Platform Concept", 2018 AIAA Information Systems - AIAA Infotech@Aerospace

The following papers are under preparation for journal submissions:

- F., D., Moncayo, H. (2021). Immunized Architecture for Multi-Agent Consensus and Increased Resilience. Journal of Swarm and Evolutionary Computation. Awaiting journal submission.
- 2. Aoun, C., F., D., Moncayo, H. (2021). Deep Reinforcement Learning with Adversary Architecture to Increase Robustness of Multi-agent Systems. Journal of Intelligent and Robotic Systems. Awaiting journal submission.